

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

IN RE APPLICATION OF: Kenichiro YOSHII et al

GAU:

SERIAL NO: NEW APPLICATION

EXAMINER:

FILED: HEREWITH

FOR: TASK ALLOCATION METHOD IN MULTIPROCESSOR SYSTEM, TASK ALLOCATION
PROGRAM PRODUCT, AND MULTIPROCESSOR SYSTEM

REQUEST FOR PRIORITY

COMMISSIONER FOR PATENTS
ALEXANDRIA, VIRGINIA 22313

SIR:

- Full benefit of the filing date of U.S. Application Serial Number , filed , is claimed pursuant to the provisions of 35 U.S.C. §120.
 - Full benefit of the filing date(s) of U.S. Provisional Application(s) is claimed pursuant to the provisions of 35 U.S.C. §119(e):
Application No. Date Filed
 - Applicants claim any right to priority from any earlier filed applications to which they may be entitled pursuant to the provisions of 35 U.S.C. §119, as noted below.

In the matter of the above-identified application for patent, notice is hereby given that:

COUNTRY

APPLICATION NUMBER

MONTH/DAY/YEAR

Certified copies of the corresponding Convention Application(s)

- are submitted herewith

will be submitted prior to payment of the Final Fee

were filed in prior application Serial No. filed

were submitted to the International Bureau in PCT Application Number
Receipt of the certified copies by the International Bureau in a timely manner under PCT Rule 17.1(a) has been acknowledged as evidenced by the attached PCT/IB/304.

(A) Application Serial No.(s) were filed in prior application Serial No. filed ; and

(B) Application Serial No.(s)

are submitted herewith

will be submitted prior to payment of the Final Fee

Respectfully Submitted,

**OBLON, SPIVAK, McCLELLAND,
MAIER & NEUSTADT P.C.**

Maryvin J. Snivak

Registration No. 24913

Registration No. 24,913

S. VIII McClelland
Registration Number 21-124

Customer Number

22850

Tel. (703) 413-3000
Fax. (703) 413-2220
(OSMMN 05/03)

0380933

日本国特許庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日 2002年11月19日
Date of Application:

出願番号 特願2002-335632
Application Number:

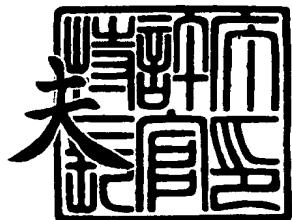
[ST. 10/C] : [JP2002-335632]

出願人 株式会社東芝
Applicant(s):

2003年 7月18日

特許庁長官
Commissioner,
Japan Patent Office

今井康



【書類名】 特許願
【整理番号】 A000205615
【提出日】 平成14年11月19日
【あて先】 特許庁長官 殿
【国際特許分類】 G06F 15/16
【発明の名称】 マルチプロセッサシステムにおけるタスク割り付け方法
、タスク割り付けプログラム及びマルチプロセッサシス
テム
【請求項の数】 27
【発明者】
【住所又は居所】 神奈川県川崎市幸区小向東芝町1番地 株式会社東芝研
究開発センター内
【氏名】 吉井 謙一郎
【発明者】
【住所又は居所】 神奈川県川崎市幸区小向東芝町1番地 株式会社東芝研
究開発センター内
【氏名】 矢野 浩邦
【発明者】
【住所又は居所】 神奈川県川崎市幸区小向東芝町1番地 株式会社東芝研
究開発センター内
【氏名】 前田 誠司
【発明者】
【住所又は居所】 神奈川県川崎市幸区小向東芝町1番地 株式会社東芝研
究開発センター内
【氏名】 金井 達徳
【特許出願人】
【識別番号】 000003078
【氏名又は名称】 株式会社 東芝

【代理人】

【識別番号】 100058479

【弁理士】

【氏名又は名称】 鈴江 武彦

【電話番号】 03-3502-3181

【選任した代理人】

【識別番号】 100084618

【弁理士】

【氏名又は名称】 村松 貞男

【選任した代理人】

【識別番号】 100068814

【弁理士】

【氏名又は名称】 坪井 淳

【選任した代理人】

【識別番号】 100092196

【弁理士】

【氏名又は名称】 橋本 良郎

【選任した代理人】

【識別番号】 100091351

【弁理士】

【氏名又は名称】 河野 哲

【選任した代理人】

【識別番号】 100088683

【弁理士】

【氏名又は名称】 中村 誠

【選任した代理人】

【識別番号】 100070437

【弁理士】

【氏名又は名称】 河井 将次

【手数料の表示】**【予納台帳番号】** 011567**【納付金額】** 21,000円**【提出物件の目録】****【物件名】** 明細書 1**【物件名】** 図面 1**【物件名】** 要約書 1**【プルーフの要否】** 要

【書類名】 明細書

【発明の名称】 マルチプロセッサシステムにおけるタスク割り付け方法、
タスク割り付けプログラム及びマルチプロセッサシステム

【特許請求の範囲】

【請求項 1】

異なる命令セットをそれぞれ有する少なくとも第1及び第2のプロセッサを含むマルチプロセッサシステムによって実行されるプログラムを構成する、前記命令セットのいずれかを用いて記述された複数のタスクを前記プロセッサに割り付けるタスク割り付け方法において、

前記タスクの中で、第1の命令セットで記述されているタスクを前記第1プロセッサに対して割り付ける第1のステップと、

前記第1プロセッサに割り付けられたタスクの少なくとも一つを対象タスクとして、該対象タスクの割り付け先を第2の命令セットを有する第2プロセッサに変更することにより前記プログラムの実行効率が向上するか否かを判定する第2のステップと、

前記実行効率が向上する場合に前記対象タスクの割り付け先を前記第2プロセッサに変更する第3のステップと

を具備するマルチプロセッサシステムにおけるタスク割り付け方法。

【請求項 2】

異なる命令セットをそれぞれ有する少なくとも第1及び第2のプロセッサを含むマルチプロセッサシステムによって実行されるプログラムを構成する、前記命令セットのいずれかを用いて記述されたプログラムモジュールとして与えられる複数のタスクを前記プロセッサに割り付けるタスク割り付け方法において、

前記タスクの中で、第1の命令セットで記述されているプログラムモジュールとして与えられるタスクを前記第1プロセッサに対して割り付ける第1のステップと、

前記第1プロセッサに割り付けられたタスクの少なくとも一つを、前記第1プロセッサが有する命令セットによって記述された第1プログラムモジュールとして与えられる対象タスクとして、該対象タスクの割り付け先を第2の命令セット

を有する第2プロセッサに変更することにより前記プログラムの実行効率が向上するか否かを判定する第2のステップと、

前記実行効率が向上する場合に前記対象タスクの割り付け先を前記第2プロセッサに変更する第3のステップと
を具備するマルチプロセッサシステムにおけるタスク割り付け方法。

【請求項3】

前記複数のタスクを前記対象タスクとして前記第1のステップの処理を行った後に前記第2及び第3のステップの処理を行う請求項1または2記載のマルチプロセッサシステムプロセッサにおけるタスク割り付け方法。

【請求項4】

前記複数のタスクから選択した一つのタスクについて前記第1、第2及び第3のステップの処理を行い、その後に前記プログラムの実行を開始する請求項1または2記載のマルチプロセッサシステムプロセッサにおけるタスク割り付け方法。

【請求項5】

前記第1のステップの処理が完了した後に前記プログラムの実行を開始し、該プログラムの実行途中で前記第2乃至第4のステップの処理を行う請求項1または2記載のマルチプロセッサシステムにおけるタスク割り付け方法。

【請求項6】

異なる命令セットをそれぞれ有する少なくとも第1及び第2プロセッサを含み、前記命令セットのいずれかを用いて記述された複数のタスクにより構成されるプログラムを実行するマルチプロセッサシステムにおいて、

前記タスクの中で、第1の命令セットで記述されているタスクを前記第1プロセッサに対して割り付ける仮割り付け部と、

前記第1プロセッサに割り付けられたタスクの少なくとも一つを対象タスクとして、該対象タスクの割り付け先を第2の命令セットを有する第2プロセッサに変更することにより前記プログラムの実行効率が向上するか否かを判定する判定部と、

前記実行効率が向上する場合に前記対象タスクの割り付け先を前記第2プロセ

ツサに変更する割り付け先変更部と
を具備するマルチプロセッサシステム。

【請求項 7】

異なる命令セットをそれぞれ有する第1及び第2プロセッサを含み、前記命令セットのいずれかを用いて記述されたプログラムモジュールとして与えられる複数のタスクにより構成されるプログラムを実行するマルチプロセッサシステムにおいて、

各タスクのプログラムモジュールをそれぞれ記述している命令セットを有する第1プロセッサに対して、該タスクをそれぞれ割り付ける仮割り付け部と、

前記各タスクの少なくとも一つを前記第1プロセッサが有する命令セットによって記述された第1プログラムモジュールとして与えられる対象タスクとして、該対象タスクの割り付け先を第2の命令セットを有する第2プロセッサに変更することにより前記プログラムの実行効率が向上するか否かを判定する判定部と、

前記実行効率が向上する場合に前記第2プロセッサが有する命令セットによって記述された第2プログラムモジュールを取得することにより、前記対象タスクの割り付け先を前記第2プロセッサに変更する割り付け先変更部と
を具備するマルチプロセッサシステム。

【請求項 8】

前記判定部は、前記各タスクのうち注目タスクの前後のタスクの割り付けられている前記第1プロセッサが該注目タスクの割り付けられている前記第1プロセッサとは異なる該注目タスクを前記対象タスクとして選定する請求項6または7記載のマルチプロセッサシステム。

【請求項 9】

前記判定部は、前記各タスクのうち注目タスクの直後のタスクに割り付けられている前記第1プロセッサが該注目タスクに割り付けられている前記第1プロセッサと同一の場合に、該注目タスクと該直後のタスクをグループ化し、該グループを一つの前記対象タスクとして取り扱う請求項6または7記載のマルチプロセッサシステム。

【請求項 10】

前記判定部は、前記対象タスクを前記第1プロセッサに割り付けた場合の前記プログラムの実行時間と前記対象タスクの割り付け先を前記第2プロセッサに変更した場合の前記プログラムの実行時間とを予測し、第2プロセッサへ割り付け先を変更した方が該予測した実行時間が短縮するか否かにより前記実行効率が向上するか否かを判定する請求項6または7記載のマルチプロセッサシステム。

【請求項11】

前記判定部は、前記対象タスクを前記第1プロセッサに割り付けた場合の単位時間当たりの処理可能データ量と前記対象タスクの割り付け先を前記第2プロセッサに変更した場合の前記タスクの単位時間当たりの処理可能データ量とを予測し、第2プロセッサへ割り付け先を変更した方が該予測した処理可能データ量が増加するか否かにより前記実行効率が向上するか否かを判定する請求項6または7記載のマルチプロセッサシステム。

【請求項12】

前記判定部は、前記対象タスクを前記第1プロセッサに割り付けた場合の単位時間当たりの処理可能データ量と前記対象タスクの割り付け先を前記第2プロセッサに変更した場合の前記タスクの単位時間当たりの処理可能データ量とを予測し、第2プロセッサへ割り付け先を変更した方が該予測した処理可能データ量が予め定められた閾値を越えて増加しているか否かにより前記実行効率が向上するか否かを判定する請求項6または7記載のマルチプロセッサシステム。

【請求項13】

前記判定部は、前記対象タスクの割り付け先を前記第2プロセッサに変更することによる該第2プロセッサの負荷を予測し、該負荷が過負荷にならないか否かにより前記実行効率が向上するか否かを判定する請求項6または7記載のマルチプロセッサシステム。

【請求項14】

前記判定部は、前記対象タスクを第1プロセッサに割り付けた場合のプロセッサ間通信データ量と前記対象タスクの割り付け先を前記第2プロセッサに変更した場合のプロセッサ間通信データ量とを予測し、第2プロセッサへ割り付け先を変更した方が該予測したプロセッサ間通信データ量が減少するか否かにより前記

実行効率が向上するか否かを判定する請求項6または7記載のマルチプロセッサシステム。

【請求項15】

前記判定部は、前記対象タスクを第1プロセッサに割り付けた場合のプログラム全体の単位時間当たりのプロセッサ間通信データ量と前記対象タスクの割り付け先を前記第2プロセッサに変更した場合のプログラム全体の単位時間当たりのプロセッサ間通信データ量とを予測し、第2プロセッサへ割り付け先を変更した方が該予測したプログラム全体の単位時間当たりのプロセッサ間通信データ量が減少するか否かにより前記実行効率が向上するか否かを判定する請求項6または7記載のマルチプロセッサシステム。

【請求項16】

前記判定部は、下記(a) (b) (c) 及び(d) の判定結果を少なくとも2つ以上組み合わせて最終的に前記実行効率が向上するか否かを判定する請求項6記載のマルチプロセッサシステム。

(a) 前記対象タスクを前記第1プロセッサに割り付けた場合の前記プログラムの実行時間と前記対象タスクの割り付け先を前記第2プロセッサに変更した場合の前記プログラムの実行時間とを予測し、第2プロセッサへ割り付け先を変更した方が該予測した実行時間が短縮するか否かにより前記実行効率が向上するか否かを判定する；

(b) 前記対象タスクを前記第1プロセッサに割り付けた場合の単位時間当たりの処理可能データ量と前記対象タスクの割り付け先を前記第2プロセッサに変更した場合の前記タスクの単位時間当たりの処理可能データ量とを予測し、第2プロセッサへ割り付け先を変更した方が該予測した処理可能データ量が増加するか否かにより前記実行効率が向上するか否かを判定する；

(c) 前記対象タスクを前記第1プロセッサに割り付けた場合の単位時間当たりの処理可能データ量と前記対象タスクの割り付け先を前記第2プロセッサに変更した場合の前記タスクの単位時間当たりの処理可能データ量とを予測し、第2プロセッサへ割り付け先を変更した方が該予測した処理可能データ量が予め定められた閾値を越えて増加しているか否かにより前記実行効率が向上するか否かを

判定する；

(d) 前記対象タスクの割り付け先を前記第2プロセッサに変更することによる該第2プロセッサの負荷を予測し、該負荷が過負荷にならないか否かにより前記実行効率が向上するか否かを判定する。

(e) 前記対象タスクを第1プロセッサに割り付けた場合のプロセッサ間通信データ量と前記対象タスクの割り付け先を前記第2プロセッサに変更した場合のプロセッサ間通信データ量とを予測し、第2プロセッサへ割り付け先を変更した方が該予測したプロセッサ間通信データ量が減少するか否かにより前記実行効率が向上するか否かを判定する。

(f) 前記対象タスクを第1プロセッサに割り付けた場合のプログラム全体の単位時間当たりのプロセッサ間通信データ量と前記対象タスクの割り付け先を前記第2プロセッサに変更した場合のプログラム全体の単位時間当たりのプロセッサ間通信データ量とを予測し、第2プロセッサへ割り付け先を変更した方が該予測したプログラム全体の単位時間当たりのプロセッサ間通信データ量が減少するか否かにより前記実行効率が向上するか否かを判定する。

【請求項17】

前記割り付け先変更部は、前記第1プログラムモジュール中の第1命令を、前記第2プログラムモジュールにおける、該第1命令と同じ処理を行う第2命令に置換することによって前記第2プログラムモジュールを取得する請求項7記載のマルチプロセッサシステム。

【請求項18】

前記割り付け先変更部は、前記第1プログラムモジュールのソースコードからコンパイラを用いて前記第2プログラムモジュールを取得する請求項7記載のマルチプロセッサシステム。

【請求項19】

前記割り付け先変更部は、前記第2プログラムモジュールをファイルシステム及びネットワークの少なくとも一方から取得する請求項7記載のマルチプロセッサシステム。

【請求項20】

前記複数のタスクの少なくとも一つは、前記複数のプロセッサの二つ以上がそれぞれ有する命令セットによって記述された複数のプログラムモジュールを含むプログラムモジュール複合体であり、

前記仮割り付け部は、前記複数のプログラムモジュール中の第1プログラムモジュールを記述している命令セットを有する一つのプロセッサを前記第1プロセッサとして、該第1プロセッサに対して前記対象タスクを割り付ける請求項7記載のマルチプロセッサシステム。

【請求項21】

前記複数のタスクの少なくとも一つは、前記複数のプロセッサの二つ以上がそれぞれ有する命令セットによって記述された複数のプログラムモジュールを含むプログラムモジュール複合体であり、

前記仮割り付け部は、前記複数のプログラムモジュール中の一つのプログラムモジュールを記述している命令セットを有する一つのプロセッサを前記第1プロセッサとして前記割り付けを行い、

前記割り付け先変更部は、前記第2プロセッサが前記複数のプログラムモジュール中の他の一つのプログラムモジュールを記述している命令セットを有する他の一つのプロセッサでない場合に、前記複数のプログラムモジュール中の任意の一つのプログラムモジュール中の第1命令を前記第2プログラムモジュールにおける、該第1命令と同じ処理を行う第2命令に置換することによって前記第2プログラムモジュールを取得する請求項7記載のマルチプロセッサシステム。

【請求項22】

前記複数のタスクの少なくとも一つは、前記複数のプロセッサの二つ以上がそれぞれ有する命令セットによって記述された複数のプログラムモジュールを含むプログラムモジュール複合体であり、

前記仮割り付け部は、前記複数のプログラムモジュール中の一つのプログラムモジュールを記述している命令セットを有する一つのプロセッサを前記第1プロセッサとして前記割り付けを行い、

前記割り付け先変更部は、前記第2プロセッサが前記複数のプログラムモジュール中の他の一つのプログラムモジュールを記述している命令セットを有する他

の一つのプロセッサでない場合に、前記複数のプログラムモジュール中の任意の一つのプログラムモジュールのソースコードからコンパイラを用いて前記第2プログラムモジュールを取得する請求項7記載のマルチプロセッサシステム。

【請求項23】

前記割り付け先変更部は、前記第2プロセッサが前記複数のプログラムモジュール中の他の一つのプログラムモジュールを記述している命令セットを有する他の一つのプロセッサでない場合に、前記プログラムモジュールをファイルシステムあるいはネットワークから取得する請求項7記載のマルチプロセッサシステム。

【請求項24】

異なる命令セットをそれぞれ有する少なくとも第1及び第2のプロセッサを含むマルチプロセッサシステムによって実行されるプログラムを構成する、前記命令セットのいずれかを用いて記述された複数のタスクを前記プロセッサに割り付ける処理をコンピュータに実行させるタスク割り付けプログラムであって、

前記タスクの中で、第1の命令セットで記述されているタスクを前記第1プロセッサに対して割り付ける第1の処理と、

前記第1プロセッサに割り付けられたタスクの少なくとも一つを対象タスクとして、該対象タスクの割り付け先を第2の命令セットを有する第2プロセッサに変更することにより前記プログラムの実行効率が向上するか否かを判定する第2の処理と、

前記実行効率が向上する場合に前記対象タスクの割り付け先を前記第2プロセッサに変更する第3の処理と

を前記コンピュータに実行させるタスク割り付けプログラム。

【請求項25】

異なる命令セットをそれぞれ有する少なくとも第1及び第2のプロセッサを含むマルチプロセッサシステムによって実行されるプログラムを構成する、前記命令セットのいずれかを用いて記述されたプログラムモジュールとして与えられる複数のタスクを前記プロセッサに割り付ける処理をコンピュータに実行せるタスク割り付けプログラムであって、

前記タスクの中で、第1の命令セットで記述されているプログラムモジュールとして与えられるタスクを前記第1プロセッサに対して割り付ける第1の処理と

前記第1プロセッサに割り付けられたタスクの少なくとも一つを、前記第1プロセッサが有する命令セットによって記述された第1プログラムモジュールとして与えられる対象タスクとして、該対象タスクの割り付け先を第2の命令セットを有する第2プロセッサに変更することにより前記プログラムの実行効率が向上するか否かを判定する第2の処理と、

前記実行効率が向上する場合に前記対象タスクの割り付け先を前記第2プロセッサに変更する第3の処理と

を前記コンピュータに実行させるタスク割り付けプログラム。

【請求項26】

前記タスク割り付けプログラムを実行させる前記コンピュータは、前記複数のプロセッサ及び前記複数のプロセッサ以外のプロセッサのうちの少なくとも一つである請求項24または25記載のタスク割り付けプログラム。

【請求項27】

前記タスク割り付けプログラムは、(a)前記複数のプロセッサの少なくとも一つのオペレーティングシステム、(b)前記複数のプロセッサ以外の少なくとも一つのプロセッサのオペレーティングシステム、(c)前記複数のプロセッサの少なくとも一つのオペレーティングシステムと前記複数のプロセッサ以外の少なくとも一つのプロセッサのオペレーティングシステム及び(d)前記マルチプロセッサシステムが実行するプログラムの少なくともいずれかの一部である請求項24または25記載のタスク割り付けプログラム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、命令セットの異なる異種プロセッサを有するマルチプロセッサシステムにおけるタスク割り付け方法、タスク割り付けプログラム及びマルチプロセッサシステムに関する。

【0002】

【従来の技術】

マルチプロセッサシステム、すなわちマルチプロセッサ計算機は、例えば「コンピュータの構成と設計 ハードウェアとソフトウェアのインターフェース 第2版（下）」、David A.Patterson, John L.Hennessy 著、成田光彰訳、日経BP社ISBN:4-8222-8057-8、第9章（非特許文献1）に記載されているように、複数のプロセッサ（CPU）によって一つのプログラムを実行する計算機である。

【0003】

各プロセッサは、バスあるいはクロスバスイッチのようなプロセッサ間結合装置によって結合される。プロセッサ間結合装置には、共有メモリ及び入出力制御装置が接続される。各プロセッサは、キャッシュメモリを持つことが多い。共有メモリを持たず、各々のプロセッサがローカルメモリを持つマルチプロセッサシステムもある。

【0004】

マルチプロセッサシステム上で実行されるプログラムの開発手法として、タスクとタスク間の依存関係でプログラムを記述する方式が広く用いられている。タスクとは、ひとまとまりの処理を行うプログラムの実行単位である。タスク間の依存関係とは、タスク間でのデータの受渡しあるいは制御の受渡しのいずれか、あるいは両方をいう。各タスクに対して、実際にプロセッサ上でそのタスクを実行するのに必要なプログラムを格納しているプログラムモジュールが存在する。このようなプログラム開発手法は、タスクのプログラムモジュールを単位としてプログラムを再利用できるという特徴を持つ。これによりプログラムの開発効率が向上し、また、過去に開発してきた数多くの優れたプログラムモジュールの資産を利用することができるという利点がある。

【0005】

タスクとタスク間の依存関係によって記述されたプログラムをマルチプロセッサシステム上で実行する際には、各タスクをどのプロセッサで実行すべきかを判断して各タスクを各プロセッサに割り付ける処理が必要がある。このタスク割り付け処理は、実行効率が高くなるように配慮して行われる。ここでいう「実行効

率が高い」とは、例えば、プログラム全体の実行時間が短いこと、単位時間当たりの処理データ量が大きいこと、各プロセッサの負荷が小さいこと、プロセッサ間通信のデータ量が小さいこと（あるいはプロセッサ間通信の回数が少ないこと）である。

【0006】

プロセッサ（CPU）は、その種類に応じて固有の命令セットを有する。命令セットとは、プロセッサが理解できる命令の集まりである。同一の命令セットを有する同種のプロセッサからなる通常のマルチプロセッサシステムとは別に、異なる命令セットを有する異種のプロセッサからなるマルチプロセッサシステム（以下、ヘテロマルチプロセッサシステムという）も存在する。ヘテロマルチプロセッサシステムは、異種プロセッサ用の複数の命令セットで記述されたプログラムモジュールをタスクとして組合せたプログラムを実行する。

【0007】

【非特許文献1】

「コンピュータの構成と設計 ハードウェアとソフトウェアのインターフェース 第2版（下）」、David A. Patterson, John L. Hennessy 著、成田光彰訳、日経BP社ISBN:4-8222-8057-8、第9章

【0008】

【発明が解決しようとする課題】

ヘテロマルチプロセッサシステムにおいても、同種のプロセッサからなる通常のマルチプロセッサシステムと同様に、プロセッサに対する各タスクの割り付けをプログラムの実行効率がより良くなるように配慮して行うことが当然に要求される。しかし、通常のマルチプロセッサシステムで用いられているタスク割り付け方法をヘテロマルチプロセッサシステムに単純に適用しても、十分なプログラム実行効率を得ることはできない。

【0009】

通常のマルチプロセッサシステムでは、各タスクを当該タスクのプログラムモジュールの記述に用いられている命令セットと同じ命令セットを有するプロセッサに割り付けている。このような通常のマルチプロセッサシステムにおけるタス

ク割り付けの手法を判断基準として、ヘテロマルチプロセッサシステムにおけるタスク割り付けを行うと、タスク間の依存関係、言い換えればタスクの実行順序の関係によって、プロセッサ間通信が頻発する。このようなプロセッサ間通信のオーバヘッドにより、ヘテロマルチプロセッサシステムではプログラムの実行効率が低下してしまうという大きな問題がある。

【0010】

本発明の目的は、命令セットの異なる異種プロセッサを有するマルチプロセッサシステムにおけるプログラムの実行効率を向上させるタスク割り付け方法、タスク割り付けプログラム及びマルチプロセッサシステムを提供することにある。

【0011】

【課題を解決するための手段】

上記の課題を解決するため、本発明ではプログラムを構成する各タスクをその記述に用いられている命令セットと同じ命令セットのプロセッサに仮割り付けした後に、割り付け先プロセッサを変更することによりプログラムの実行効率が向上するか否かを判定し、その判定結果に従って必要な場合に対象タスクの割り付け先を変更して本割り付けを行う。

【0012】

すなわち、本発明の一つの態様では、異なる命令セットをそれぞれ有する少なくとも第1及び第2のプロセッサを含むマルチプロセッサシステムによって実行されるプログラムを構成する、前記命令セットのいずれかを用いて記述された複数のタスクを前記プロセッサに割り付ける際、まず前記タスクの中で第1の命令セットで記述されているタスクを第1プロセッサに対して割り付ける。次に、第1プロセッサに割り付けられたタスクの少なくとも一つを対象タスクとして、該対象タスクの割り付け先を第2の命令セットを有する第2プロセッサに変更することによりプログラムの実行効率が向上するか否かを判定する。この判定結果に従って、実行効率が向上する場合に前記対象タスクの割り付け先を第2プロセッサに変更する。

【0013】

より具体的な態様では、マルチプロセッサシステムによって実行されるプログ

ラムを構成する各タスクは、各プロセッサが有する異なる命令セットのいずれかでそれぞれ記述されたプログラムモジュールとして与えられる。そして、対象タスクの割り付け先を第1プロセッサから第2プロセッサに変更することによりプログラムの実行効率が向上する場合には、第2プロセッサが有する命令セットによって記述されたプログラムモジュールを取得することにより、対象タスクの割り付け先を第2プロセッサに変更する。

【0014】

本発明によると、異なる命令セットをそれぞれ有する少なくとも第1及び第2のプロセッサを含むマルチプロセッサシステムによって実行されるプログラムを構成する、前記命令セットのいずれかを用いて記述された複数のタスクを前記プロセッサに割り付ける処理をコンピュータに実行させるタスク割り付けプログラムであって、前記タスクの中で、第1の命令セットで記述されているタスクを前記第1プロセッサに対して割り付ける第1の処理と、前記第1プロセッサに割り付けられたタスクの少なくとも一つを対象タスクとして、該対象タスクの割り付け先を第2の命令セットを有する第2プロセッサに変更することにより前記プログラムの実行効率が向上するか否かを判定する第2の処理と、前記実行効率が向上する場合に前記対象タスクの割り付け先を前記第2プロセッサに変更する第3の処理とを前記コンピュータに実行させるタスク割り付けプログラムが提供される。

【0015】

さらに、本発明によると異なる命令セットをそれぞれ有する少なくとも第1及び第2のプロセッサを含むマルチプロセッサシステムによって実行されるプログラムを構成する、前記命令セットのいずれかを用いて記述されたプログラムモジュールとして与えられる複数のタスクを前記プロセッサに割り付ける処理をコンピュータに実行させるタスク割り付けプログラムであって、

前記タスクの中で、第1の命令セットで記述されているプログラムモジュールとして与えられるタスクを前記第1プロセッサに対して割り付ける第1の処理と、前記第1プロセッサに割り付けられたタスクの少なくとも一つを、前記第1プロセッサが有する命令セットによって記述された第1プログラムモジュールとし

て与えられる対象タスクとして、該対象タスクの割り付け先を第2の命令セットを有する第2プロセッサに変更することにより前記プログラムの実行効率が向上するか否かを判定する第2の処理と、前記実行効率が向上する場合に前記対象タスクの割り付け先を前記第2プロセッサに変更する第3の処理とを前記コンピュータに実行させるタスク割り付けプログラムが提供される。

【0016】

ここで、前記タスク割り付けプログラムを実行させる前記コンピュータは、例えば前記複数のプロセッサ及び前記複数のプロセッサ以外のプロセッサのうちの少なくとも一つである。

【0017】

前記タスク割り付けプログラムは、具体的には例えば（a）前記複数のプロセッサの少なくとも一つのオペレーティングシステム、（b）前記複数のプロセッサ以外の少なくとも一つのプロセッサのオペレーティングシステム、（c）前記複数のプロセッサの少なくとも一つのオペレーティングシステムと前記複数のプロセッサ以外の少なくとも一つのプロセッサのオペレーティングシステム及び（d）前記マルチプロセッサシステムが実行するプログラムの少なくともいずれかの一部として構成される。

【0018】

このように本発明によると、命令セットが異なる複数種類のプロセッサから構成されるヘテロマルチプロセッサシステムにおいて、記述に用いられている命令セットが異なる複数のタスク群を実行する際に、命令セットが異なるプロセッサへ割り付けた方がよいタスクの選定と割り付けの変更が実現でき、これによりシステム全体のプログラム実行効率が向上する。

【0019】

【発明の実施の形態】

以下、図面を参照しながら発明の実施の形態を説明する。

（マルチプロセッサシステムの全体構成）

図1に、本発明の一実施形態に係るマルチプロセッサシステムの基本的な構成例を示す。このシステムは、いわゆるヘテロマルチプロセッサシステムであり、

命令セットA, B及びCをそれぞれ有する複数のプロセッサ1～3と、共有メモリ4及び入出力制御装置5がバスやクロスバスイッチ等のプロセッサ間結合装置7によって接続されている。入出力制御装置5には大容量記憶装置、例えばディスク装置6が接続されている。プロセッサ間結合装置7には、さらに図1では概念的に示したタスク割り付けシステム8が結合される。

【0020】

図1中には示されていないが、プロセッサ1～3はキャッシュやローカルメモリを持っていてもよい。マルチプロセッサシステムは、共用メモリを持たなくともよい。図1では、3個のプロセッサ1～3が示されているが、マルチプロセッサシステムは2個あるいは4個以上のプロセッサを含んでいてもよい。ヘテロマルチプロセッサシステムに含まれる複数のプロセッサは、全て異なる命令セットを使用している必要はなく、2つまたはそれ以上のプロセッサが同一の命令セットを有する構成でもよい。要するに、ヘテロマルチプロセッサシステムは、異なる命令セットを有する少なくとも2つの異種プロセッサを含んでいればよい。

【0021】

マルチプロセッサシステムが実行するプログラムを構成する各々のタスクに対して、実際にプロセッサ1～3上でタスクを実行するために必要なプログラムを格納しているプログラムモジュールは、入出力制御装置5の前に接続されているディスク装置6あるいは共有メモリ4に格納される。共有メモリがなく、プロセッサ内のローカルメモリが存在するマルチプロセッサシステムでは、当該ローカルメモリにプログラムモジュールが格納される。プログラムモジュールは、当該タスクを実行するために必要な命令が特定の命令セットで記述されている。

【0022】

(タスク割り付けシステムの実装例)

タスク割り付けシステム8は、マルチプロセッサシステムが実行するプログラムの各タスクをプロセッサ1～3に適切に割り付けるものであり、具体的にはプログラム（以下、タスク割り付けプログラムという）として実装される。タスク割り付けプログラムは、タスク割り付けのみを行う専用のプログラムであってもよいし、オペレーティングシステムの一部であったり、オペレーティングシス

ムとは別のメインプログラムであってもよい。図2～図5に、タスク割り付けプログラムの実装例を示す。

【0023】

図2の例では、特定のプロセッサ1上で動作するオペレーティングシステム(OS)11の一部としてタスク割り付けプログラム12が存在している。タスク割り付けプログラム12は、これが存在するオペレーティングシステム11が動作しているプロセッサ1を含めた全てのプロセッサ1～3に対するタスク割り付け処理を司る。

【0024】

図3の例では、マルチプロセッサシステムに含まれる全てのプロセッサ1～3上で動作するオペレーティングシステム11の一部として、タスク割り付けプログラム12が存在している。図3のシステムでのタスク割り付け処理の様子は、二つ考えられる。一つの様子では、各々のプロセッサ1～3上で動作しているオペレーティングシステム11の一部であるタスク割り付けプログラム12が完全に対等の関係で協調してタスク割り付け処理を行う。

【0025】

図3におけるタスク割り付け処理の他の様子では、特定のプロセッサ上で動作しているオペレーティングシステムの一部であるタスク割り付けプログラムをメインのプログラムとし、他のプロセッサで動作しているオペレーティングシステムの一部であるタスク割り付けプログラムをサブのプログラムとして、これらメイン及びサブのプログラムが協調してタスク割り付け処理を行う。

【0026】

図4の例では、マルチプロセッサシステムを構成する主たるプロセッサ1～3とは別に管理用プロセッサ9が設けられ、この管理用プロセッサ9上で動作するオペレーティングシステム13の一部として、タスク割り付けプログラム12が存在する。管理用プロセッサ9には、マルチプロセッサシステムが実行するプログラムのタスクは割り付けられない。

【0027】

図5は、図3と図4を組み合わせた例であり、プロセッサ1～3上で動作して

いるオペレーティングシステム11の一部及び管理用プロセッサ9上で動作しているオペレーティングシステム13の一部であるタスク割り付けプログラム12のうち、後者がタスク割り付けプログラムのメインのプログラムとして動作し、前者はサブのプログラムとしてメインのプログラムと協調してタスクの割り付け処理を行う。

【0028】

図2～図5では、上述したようにタスク割り付けプログラムがオペレーティングシステムの一部である例について述べたが、タスク割り付けプログラムがメインプログラムの一部であったり、タスク割り付けのみを行う専用のプログラムである場合にも、タスク割り付けプログラムを同様に配置することが可能である。

【0029】

(マルチプロセッサシステムが実行するプログラムについて)

本実施形態のマルチプロセッサシステムが実行するプログラムは、図6に示されるように複数のタスクT1～T6とタスクT1～T6間の依存関係で記述される。前述したように、タスクT1～T6はひとまとめの処理を行うプログラムの実行単位である。タスクT1～T6間の依存関係は、タスクT1～T6間でのデータの受渡しあるいは制御の受渡しのいずれか、あるいは両方であり、図6では矢印によってタスクからタスクへのデータまたは制御の受け渡しが示されている。タスクのプログラムモジュールを実行した時には、この矢印に従ってタスク間でデータ転送が行われる。

【0030】

(プログラムのタスク実行例)

図7 (a) (b) (c) に、タスクの実行の様子の種々の例を示す。

図7 (a) の例は、一入力一出力のタスクの実行の様子を示している。タスクの実行は、まず入力元となるタスクから処理に必要なデータを受信し、次にそのデータに対して処理を行い、最後に出力先となるタスクに対してデータを送信する、という3つの段階からなる。

図7 (b) には、2入力2出力のタスクの実行の様子を示す。この例では、全ての入力元タスクからデータを受け取ってから、そのデータに対して処理を行

、最後に出力先にデータを送信する。

図7（c）は、図7（a）（b）は異なり、入力データは一度に全てを与えられるわけではなく、断続的に入力元となるタスクから与えられ、例えばある時間単位に受信したデータに対して処理を行い、その処理結果のデータを逐次出力先のタスクに送信するというタスク実行の様子を示している。

このようなタスク実行に伴うタスク間のデータ送受信にかかるコストは、マルチプロセッサシステムの構成にも大きく依存するが、一般的に比較的高い。

【0031】

さらに、図1に示したような共有メモリ4を有するマルチプロセッサシステムでは、データを送信するタスクとデータを受信するタスクが同じプロセッサに割り付けられているか、異なるプロセッサに割り付けられているかに関わらず、データの送信は共有メモリ4への書き込み、そしてデータの受信は共有メモリ4からの読み出しで実現される。一般的に、共有メモリ4に対する書き込み／読み出しのコストも高い。

【0032】

一方、プロセッサがキャッシュを有するマルチプロセッサシステムでは、もしデータの送信と受信をするタスクが同じプロセッサに割り付けされていた場合には、それらのタスク間のデータ送受信はプロセッサ内のキャッシュを介して行われる。通常、キャッシュへのアクセスは共有メモリへのアクセスに比べて高速であるため、タスクから見ると、処理結果のデータの送信や処理に必要なデータの受信がキャッシュへの読み書きによって行われる分、見かけ上データ送受信のコストは下っている。しかし、キャッシュはその内容についてメモリとの整合性を保つ必要があるため、実際にはやはりメモリへの書き込みが発生する。

【0033】

逆に、データを送信するタスクと受信をするタスクが異なるプロセッサに割り付けされていた場合には、キャッシュの仕組みによっても異なるが、データの送信は共有メモリへの書き込みによって、データの受信は共有メモリからの読み出しによって行われることにより、タスク間のデータ送受信が実現される。このような共有メモリを介してのデータ送受信も、やはりコストが高い。

【0034】

次に、プロセッサがローカルメモリを有するマルチプロセッサシステムでは、もしデータの送信と受信をするタスクが同じプロセッサに割り付けされていた場合は、それらのタスクの間ではプロセッサ内のローカルメモリを利用したデータの送受信が行われる。ローカルメモリへのアクセスは通常、共有メモリへのアクセスに比べ高速である。しかし、データの送信と受信をするタスクが異なるプロセッサに割り付けされていた場合には、送信元のタスクが割り付けられているプロセッサのローカルメモリから送信先のタスクが割り付けられているプロセッサ内のローカルメモリへのデータ転送によって、タスク間のデータ送受信が実現される。このローカルメモリ間の通信は通常、共有メモリへのアクセスと同様にコストが高い。

【0035】

このようにマルチプロセッサシステムにおいては、プロセッサ間通信に伴うコストが高いため、プロセッサ間通信を十分に考慮してタスクをプロセッサに割り付ける必要がある。

【0036】

従来のタスク割り付けにおいては、各タスクの実行に必要なプログラムを格納しているプログラムモジュールの記述に使用されている命令セットと同一命令セットを採用しているプロセッサへタスクを割り付けている。このような割り付け方法を本実施形態のようなヘテロマルチプロセッサシステムに適用すると、頻繁にタスク間のデータ通信がプロセッサ間で行われることにより、実行効率が悪くなってしまう。

【0037】

この問題を緩和するため、本実施形態では従来の各タスクの実行に必要なプログラムを格納しているプログラムモジュールの記述に使用されている命令セットと同一命令セットを採用しているプロセッサへの割り当てを「仮割り当て」と位置付け、この「仮割り当て」の後に、プログラムの実行効率がより高くなるように各タスクのプロセッサへの割り付けを最適化する。

【0038】

(タスク割り付けシステムの詳細)

次に、タスク割り付けシステム8について詳しく述べる。図8は、図1中に示したタスク割り付けシステム8の構成例を示している。前述のように、タスク割り付けシステム8は専用のタスク割り付けプログラム、オペレーティングシステムの一部、あるいはオペレーティングシステムとは別のメインプログラムで実現されるが、図8では分かることなくためタスク割り付け部8の機能をブロック図で表している。

【0039】

図8において、タスク仮割り付け部21は上述した仮割り付け、すなわち各タスクの実行に必要なプログラムを格納しているプログラムモジュールの記述に使用されている命令セットと同一命令セットを採用しているプロセッサへのタスクの割り付けを行う。各タスクの仮割り付けに関する情報は、例えば図1中のディスク装置6または共有メモリ4の一部である仮割り付けタスク保持部22に保持されており、仮割り付けタスク読出部23により読み出される。

【0040】

仮割り付けタスク読出部23によって読み出された情報は、最適化対象タスク判定部24に入力される。最適化対象タスク判定部24では、マルチプロセッサシステムが実行する各プログラムを構成する全タスクについて、最適化によって割り付け先を変更した方がよいかどうかが判定される。各タスクのうち最適化対象と判定されたタスクに対して、実際に最適化によるプロセッサへの割り付けの変更を行うかどうかが、最適化実行判定部25によって判定される。

【0041】

最適化によるプロセッサへの割り付け先の変更を実行することになったタスクに対して、最適化実行部26により実際に割り付け先の変更処理が行われる。割り付け先の変更を行ったかどうかに関わらず、全てのタスクについてその最終的な割り付け結果の情報が割り付けタスク書込部27によって、例えば図1中のディスク装置6または共有メモリ4の一部である割り付けタスク保持部28に書き込まれる。

【0042】

図9に示されるように、最適化実行判定部25はプログラムの実行効率を予測する手段として、例えば実行時間予測部31、単位時間処理可能データ量予測部32、プロセッサ負荷予測部33及びプロセッサ間通信データ量予測部34を有する。予測方法選択部35によって、いずれか一つまたは複数の予測部が実行効率判定のために選択される。

【0043】

ここで、実行時間予測部31は対象タスクを仮割り付け先に割り付けた場合と割り付け先を変更した場合のタスクの実行時間とを予測する。単位時間処理可能データ量予測部32は、対象タスクを仮割り付け先に割り付けた場合と割り付け先を変更することによるプログラムの単位時間当たりの処理可能データ量を予測する。プロセッサ負荷判定部33は、対象タスクの割り付け先を変更することによる割り付け先プロセッサの負荷を予測する。プロセッサ間通信データ量予測部34は、対象タスクを仮割り付け先に割り付けた場合と割り付け先を変更した場合のプログラムのプロセッサ間通信データ量とを予測する。

【0044】

実行効率判定部36では、予測方法選択部35により選択された予測部の予測結果に基づいてプログラムの実行効率を判定する。具体的には、実行時間予測部31は（a）実行時間予測部31の予測した実行時間が割り付け先の変更によって短縮するか否か、（b）単位時間処理可能データ量予測部32が予測した処理可能データ量が割り付け先の変更によって増加するか否か、または予測した処理可能データ量が割り付け先の変更によって予め定められた閾値を越えて増加するか否か、（c）プロセッサ負荷予測部33が予測したプロセッサの負荷が過負荷にならないか否か、（d）プロセッサ間通信データ量予測部34が予測したプロセッサ間通信データ量が割り付け先の変更によって減少するか否かにより、タスク割り付け先の変更によってプログラムの実行効率が向上するか否かの判定を行う。

【0045】

予測方法選択部35により選択された予測部が複数の場合には、実行効率判定部36はそれら複数の予測結果を総合的に判断して、実行効率が向上するか否か

を最終的に判定する。これらの実行効率判定の具体的な手法については、後に詳しく説明する。

【0046】

こうして実行効率判定部36によって「タスク割り付け先の変更によりプログラムの実行効率が向上する」と判定されたタスクに対しては、割り付け先プロセッサ決定部37により新たな割り付け先プロセッサが決定される。「タスク割り付け先を変更してもプログラムの実行効率が向上しない」と判定されたタスクについては、仮割り付け先のプロセッサが最終的な割り付け先プロセッサとして決定される。

【0047】

図10は、複数の異種プロセッサ用の命令セットで記述されたプログラムモジュールをタスクT1～T9として組合せたプログラムの例である。各タスクT1～T9のプログラムモジュールを記述している命令セットは、括弧内のアルファベットA, B, Cで示されている。すなわち、図10のプログラムは命令セットAで記述されたプログラムモジュールを持つタスクT1, T5, T9と、命令セットBで記述されたプログラムモジュールを持つタスクT2, T6、及び命令セットCで記述されたプログラムモジュールを持つタスクT3, T4, T7, T8から構成される。

【0048】

従来のタスク割り付け方法に従うと、図10に示したプログラム中の各タスクは図11に示されるように、そのプログラムモジュールを記述している命令セットを有するプロセッサに割り付けられる。すなわち、タスクT1, T5, T9は命令セットAを有するプロセッサ1に、タスクT2, T6は命令セットBを有するプロセッサ2に、そしてタスクT3, T4, T7, T8は命令セットCを有するプロセッサ3にそれぞれ割り付けられる。

【0049】

これに対して、本実施形態では前述したように図11のタスク割り付けを仮割り付けと位置付け、この仮割り付け後の最適化により例えば図12に示すように割り付け先のプロセッサを変更することができる。これによりプロセッサ間での

通信が必要なタスク間のデータ送受信の回数は、図11に示す7回から図12に示す2回へと大きく減少する。すなわち、プロセッサ間通信によるオーバヘッドが減少し、プログラムの実行効率が大きく改善される。

【0050】

(タスク割り付け処理手順1)

次に、本実施形態に基づくタスク割り付けの処理手順について、フローチャートを用いて説明する。図13は、本実施形態におけるタスク割り付け処理の一例の基本的な流れを示している。図13に示した手順をタスク割り付け処理手順1とする。

【0051】

まず、図8中のタスク仮割り付け部21によって、プログラムを構成する全タスクを各プロセッサに対して仮割り付けする（ステップS11）。各タスクの仮割り付けに関する情報は、図8中の仮割り付けタスク保持部22に保持される。次に、仮割り付けタスク保持部22から仮割り付けタスク読出部23により仮割り付けに関する情報が読み出され、最適化対象タスク判定部24に送られる。

【0052】

最適化対象タスク判定部24では、プログラムを構成する全タスクのうち、割り付け先プロセッサの変更により実行効率が向上する可能性のあるタスク（最適化対象タスク）と判定された対象タスクについて、最適化実行判定部25において割り付け先プロセッサの変更によりプログラム実行効率が向上するか否かを判定する（ステップS12）。

【0053】

ここで、ステップS12において割り付け先プロセッサの変更によりプログラム実行効率が向上しないと判定されたタスクに対しては、ステップS11での仮割り付け先のプロセッサを最終的な割り付け先プロセッサとして決定して処理を終了する。一方、割り付け先プロセッサの変更によりプログラム実行効率が向上する場合には、新たな割り付け先プロセッサを決定する。

【0054】

次に、割り付け先プロセッサの変更によりプログラム実行効率が向上すると判

定された対象タスクに対して、決定された新たなプロセッサに割り付け先を変更する（ステップS13）。割り付け先プロセッサの変更とは、具体的には対象タスクについて新たな割り付け先プロセッサが有する命令セットで記述されたプログラムモジュールを取得することである。

【0055】

図13に示した処理が終了すると、プログラムを構成する全タスクがそれぞれ適切なプロセッサに割り付けられる。これによって、マルチプロセッサシステムは当該プログラムを効率よく実行することができる。

【0056】

次に、図13の各ステップS11～S13の処理について詳しく説明する。

図14には、図13中のステップS11の処理の詳細を示す。割り付けるべき対象タスクのプログラムモジュールを記述している命令セットが何であるかを判断し（ステップS101）、その命令セットを有するプロセッサに対して対象タスクを割り付ける（ステップS102）。図10に示したプログラムを例にとると、このタスク仮割り付けでは、図10のプログラム中の各タスクを図11に示したように各プロセッサに割り付けることになる。

【0057】

図15は、図13中のステップS12の詳細な処理を示すフローチャートである。図15では一つの対象タスクに対する処理を記述しているが、実際にはプログラムを構成する全てのタスクに対して同様の処理を行う。この処理を同じ対象タスクに対して複数回にわたり適用することもできる。例えば、一度プログラムを構成する全てのタスクについて図15の処理を行い、幾つかのタスクについて最適化による割り付けの変更を行った後に、その結果のタスク群に対して再度、同じ処理を施してもよい。このようにすることで、よりよい最適化の結果が得られる可能性がある。

【0058】

まず、仮割り付けタスク読出部23によって読み出されたタスク仮割り付けに関する情報は、最適化対象タスク判定部24に送られる。最適化対象タスク判定部24は、ステップS11で仮割り付けされたタスクに対して、現在注目してい

る対象タスクの直前または直後のタスクが、対象タスクが仮割り付けされているプロセッサとは命令セットが異なるプロセッサに割り付けられているか否かを判定する（ステップS201）。

【0059】

ここで、例えば図10のプログラムにおけるタスクT1, T2, T4, T5のような、直前のタスクが存在しない対象タスクについては、「直前のタスク」として仮想的なタスクを定義する。仮想的なタスクとは、例えば予想実行時間が0で、かつ当該対象タスクへ送信するデータも0、さらにプロセッサの負荷には全く影響を及ぼさないタスクである。さらに、図10中のタスクT9のような、直後のタスクが存在しない対象タスクについても、同様に「直後のタスク」を定義する。

【0060】

ステップS201の判定結果がYESであれば、対象タスクに関する情報を最適化対象タスクとして最適化実行判定部24へ渡し、ステップS202の処理を行う。一方、ステップS201の判定結果がNO、すなわち対象タスクの直前及び直後の両タスクが対象タスクと同じプロセッサに仮割り付けされている場合には、対象タスクの割り付け先プロセッサを変更する必要はない、言い換えれば割り付け先プロセッサを変更しても実行効率は向上しないので、その旨の判定結果を割り付けタスク書込部27へ渡し、仮割り付けタスクに関する情報を割り付けタスク保持部28に書き込んで処理を終了する。

【0061】

ステップS202では、ステップS201により最適化対象タスクと判定されたタスクを既に当該タスクが仮割り付けされたプロセッサと割り付け変更先候補のプロセッサにそれぞれ割り付けた場合の各々におけるプログラムの実行効率を最適化実行判定部25で予測する。ここで、割り付け変更先候補のプロセッサとは、現在注目している最適化対象タスクの仮割り付け先プロセッサと異なるプロセッサが仮割り付け先となっている、当該タスクの直前及び直後のタスクが割り付けられているプロセッサの全てである。

【0062】

最適化実行判定部25は、引き続き最適化対象タスクの割り付け先を割り付け変更先候補のプロセッサにした方がプログラムの実行効率が向上するかどうかを判定する（ステップS203）。ここで、ステップS203の判定結果がYESであれば、最適化実行判定部25は割り付け変更先候補プロセッサを割り付け先プロセッサと決定し（ステップS204）、決定された割り付け変更先プロセッサに割り付け先を変更する旨の印を最適化対象タスクに付け（ステップS205）、処理を終了する。ステップS203の判定結果がNOであれば、そのまま処理を終了する。

【0063】

（タスクのグループ化について）

プログラムが図10に示したような簡単なものでなく、例えばタスク数が多く規模の大きなプログラム、タスク間依存関係の複雑なプログラム、あるいはタスク数が多くかつタスク間依存関係も複雑なプログラムでは、最適化対象タスク判定部24及び最適化実行判定部25での処理が複雑になることが考えられる。

【0064】

図16は、こうした複雑なプログラムに対するタスク割り付け処理を簡単にするため、プログラムを構成する各タスクをグループ化する処理を示している。この処理は、例えば図15のステップS201の前処理として配置される。このようなタスクのグループ化によって、タスクの仮割り付け図を簡略化し、図15の処理を簡略化することができる。図16では、一つのタスクに関する処理を例示しているが、実際にはプログラムを構成する全てのタスクについて同様の処理を行う。

【0065】

図16の処理の流れを説明すると、まず注目している対象タスクの直後にタスクがあるかどうかを判定する（ステップS211）。ステップS211の判定結果がYESであれば、直後のタスクが全て対象タスクと同じプロセッサに仮割り付けされているかどうかを判定する（ステップS212）。

【0066】

ステップS212の判定結果がYESであれば、対象タスクのみが先行タスク

となっている直後のタスクを選択する（ステップS213）。こうして選択したタスクと対象タスクをグループ化し（ステップS214）、このグループを一つの対象タスクとして取り扱い、図15のステップS201に渡す。このようなグループ化により、複雑なプログラムについてもタスク割り付け処理を容易にすることができる。

【0067】

（最適化実行判定）

次に、図13中の判定ステップS12の処理、特に図15中に示したステップS202及びS203の処理について説明する。この処理は、図9に詳細な構成を示した図8中の最適化実行判定部25によって、以下に列挙する幾つかの実行効率判定基準を単独で用いるか、あるいは幾つか組み合わせて行われる。

【0068】

〔実行効率判定基準1〕

割り付け先プロセッサを変更することにより、プログラム実行時間（タスクの実行に要する時間）が短くなるかどうかを判定する。

タスクの実行に要する時間は、タスクの実行に必要なプログラムを格納するプログラムモジュールに記述してある命令列から見積もることができ、割り当て変更先の候補となっているプロセッサでのタスクの実行に要する時間も同様にして見積もることができる。

【0069】

この実行効率判定基準1によれば、プロセッサに割り付けるべき対象タスクに対して、仮割り付けされているプロセッサでの実行に必要な予測実行時間よりも、割り付け変更先候補となっているいずれかのプロセッサでの実行に必要な予測実行時間の方が短ければ、その対象タスクは最適化対象タスク、すなわち最適化により割り付け先プロセッサを変更すべきタスクであると判定する。

【0070】

複数の割り付け変更先候補プロセッサでの予測実行時間が仮割り付け先プロセッサでの予測実行時間よりも短い場合も考えられる。この場合には、最も予測実行時間の短いプロセッサを割り付け変更先として選定するか、もしくは実行

効率判定基準1では複数のプロセッサを割り付け変更先候補として選定し、最終的な割り付け変更先プロセッサの選定は、他の実行効率判定基準による判定に委ねるという方法もある。

【0071】

[実行効率判定基準2]

割り付け先プロセッサを変更することにより、タスクが単位時間内に処理できるデータ量が多くなるかどうかを判定する。

タスクが単位時間内に処理できるデータ量とは、すなわちタスクが単位時間に先行タスクから受信できるデータ量である。単位時間内に先行タスクからタスク間通信により受信できるデータ量は、現在注目している対象タスクと各先行タスクが同一プロセッサに仮割り付けされているか、異なるプロセッサに仮割り付けされているかの違いによって影響を受ける。これは異なるプロセッサ間の通信は、同一プロセッサ内での通信に比べて非常にコストが高いからである。

【0072】

実行効率判定基準2によると、まず単位時間内に全ての先行タスクとのタスク間通信によって受信できるデータ量について、仮割り付け先となっているプロセッサでの場合と、各割り付け変更先候補となっているプロセッサについてそれぞれ予測する。

【0073】

ここで、もし現在の仮割り付け先プロセッサにおいて単位時間内に受信するとのできるデータ量よりも、割り付け変更先候補となっているプロセッサのいずれかに割り付けを変更した場合の方が、単位時間内に受信することのできるデータ量が増加していれば、現在注目しているタスクに対して当該割り付け変更先候補プロセッサへ割り付け先プロセッサを変更するべきと判断する。

【0074】

複数の割り付け変更先候補プロセッサに割り付けを変更した場合に、現在注目している対象タスクが単位時間に受信できるデータ量が、現在注目しているタスクが仮割り当てされているプロセッサ上で単位時間内に受信できるデータ量よりも多い場合も考えられる。この場合には、現在注目している対象タスクが単位時

間内に最も多いデータ量を受信することのできるプロセッサを割り付け変更先として選定する。

【0075】

複数の割り付け変更先候補プロセッサでの対象タスクが単位時間内に受信できるデータ量が同じで、かつ仮割り当てされているプロセッサ上での単位時間内に受信することのできるデータ量よりも多い場合も含めて、実行効率判定基準2では複数のプロセッサを割り付け変更先候補として選定し、最終的な割り付け変更先プロセッサの選定は、他の実行効率判定基準による判定に委ねるという方法もある。

【0076】

[実行効率判定基準3]

割り付け先プロセッサを変更することにより、タスクが単位時間に処理できるデータ量が予め設定された閾値よりも多くなるかどうかを判定する。

これは実行効率判定基準2と基本的に同様の判定であるが、現在注目している対象タスクが単位時間に受信できるデータ量を仮割り付け先プロセッサと割り付け変更先候補プロセッサについて比較する際に、予め単位時間内に受信できるデータ量について、開始する前に設定された静的な閾値、もしくは選定中に動的に設定された動的な閾値よりを導入する。

【0077】

ここで、もし仮割り付け先プロセッサよりも割り付け変更先候補プロセッサのいずれかの方が、現在注目している対象タスクが単位時間に受信できるデータ量が多く、かつ閾値よりも大きい場合には、対象タスクに対して当該割り付け変更先候補プロセッサへ割り付け先プロセッサの変更を行うべきと判定する。

【0078】

[実行効率判定基準4]

割り付け先プロセッサを変更することにより、割り付け変更先プロセッサが過負荷にならないかどうかを判定する。

割り付け先プロセッサを仮割り付け先プロセッサから変更したとしても、割り付け変更先となったプロセッサが過負荷になってしまっては、プログラム全体の

実行効率の改善にはならない。

【0079】

そこで、現在注目しているタスクを仮割り付け先プロセッサにそのまま割り付けた場合の全プロセッサの負荷を予測する。さらに、現在注目しているタスクを割り付け変更先候補のプロセッサのいずれかに割り当て先を変更した場合の全プロセッサの負荷をそれぞれ予測する。そして、割り付けを変更した場合に、割り付け変更先候補プロセッサが過負荷になっていなければ、最適化による割り付け先プロセッサの変更を行うべきと判断する。

【0080】

割り付け先を変更しても、全プロセッサの予測した負荷が過負荷にならないような複数の割り付け変更先候補プロセッサが存在する場合もある。このような場合には、最も負荷の変動が少ない割り付け変更先候補プロセッサを選定したり、現在注目している対象タスクの割り付け先を変更しても最も負荷が少ない割り付け変更先候補プロセッサを選定するなどの方法をとることができる。さらには、この実行効率判定基準4では複数のプロセッサを割り付け先候補として選定し、最終的な割り当て変更先プロセッサの選定は他の実行効率判定基準による判定に委ねるという方法もある。

【0081】

[実行効率判定基準5]

割り付け先プロセッサを変更することにより、プログラム全体でのプロセッサ間の通信データ量が小さくなるかどうかを判定する。

マルチプロセッサシステムにおけるプログラムの実行効率改善の鍵は、やはりプロセッサ間通信のデータ量である。この点に着目して、仮割り付け先プロセッサと割り付け変更先候補のプロセッサとで、プログラム全体におけるプロセッサ間通信によって転送されるデータ量が削減されるかどうかを判定基準とする。

【0082】

具体的には、現在注目している対象タスクの割り付け先プロセッサを変更しない場合と、割り付け変更先候補プロセッサのいずれかに割り付け先を変更した場合について、プログラム全体でのプロセッサ間通信によって転送されるデータ量

を見積もる。もし、いずれかの割り付け変更先プロセッサに割り付け先を変更した方が、割り付け先変更前よりもプログラム全体におけるプロセッサ間通信により転送されるデータ量が減少する場合には、当該割り付け変更先候補プロセッサへ割り付けを変更した方がよいと判断する。

【0083】

対象タスクを複数の割り付け変更先候補プロセッサに割り付け先を変更した場合のプログラム全体でのプロセッサ間通信によって転送されると予想されるデータ量が、対象タスクが仮割り付けされているプロセッサに対してそのまま割り付けた際にプログラム全体でのプロセッサ間通信によって転送されると予想されるデータ量よりも少ないと考えられる。この場合には、プログラム全体で最もプロセッサ間通信によって転送されるデータ量が少ない割り付け変更先候補プロセッサを割り付け変更先プロセッサとして選定する。あるいは、この実行効率判定基準5では複数のプロセッサを割り付け先候補として選定し、最終的な割り当て変更先プロセッサの選定は他の実行効率判定基準による判定に委ねるという方法もある。

【0084】

[実行効率判定基準6]

割り付け先プロセッサを変更することにより、プログラム全体での単位時間のプロセッサ間の通信データ量が小さくなるかどうか。

これは実行効率判定基準5と基本的に同様の判定基準であるが、仮割り付け先プロセッサと各割り付け変更先候補プロセッサについてプロセッサ間転送データ量を単位時間当たりについて見積もる。割り付け変更先候補プロセッサのいずれかに割り付けを変更した場合のプログラム全体で単位時間内にプロセッサ間で転送されるデータ量が、仮割り付け先プロセッサにおけるプログラム全体で単位時間にプロセッサ間で転送されるデータ量よりも少なければ、当該割り付け変更先候補プロセッサへ割り付けを変更した方がよいと判定する。

【0085】

(タスク割り付け処理の具体例)

次に、上述したタスク割り付け処理の手順を具体的なプログラムの例を用いて

説明する。

以下の説明では、図11のように仮割り付けされた図10のプログラムの各タスクT1～T9の割り付け先を最適化する手順について詳細に述べる。図10のプログラムは、命令セットAで記述されたプログラムモジュールを持つタスクT1, T5, T9と、命令セットBで記述されたプログラムモジュールを持つタスクT2, T6、及び命令セットCで記述されたプログラムモジュールを持つタスクT3, T4, T7, T8から構成される。

【0086】

図10のプログラムに対する図11に示した仮割り付け結果においては、タスクT1, T5, T9は命令セットAを有するプロセッサ1に、タスクT2, T6は命令セットBを有するプロセッサ2に、そしてタスクT3, T4, T7, T8は命令セットCを有するプロセッサ3にそれぞれ割り付けられている。

【0087】

ここで説明するタスク割り付け処理の例では、割り付け先プロセッサを変更するかどうか、さらに割り付け先をどのプロセッサに変更するかどうかの判定に、先の実行効率判定基準1及び5のみを使用することとする。また、実行効率判定基準1より実行効率判定基準5の方が優先度が高いと仮定する。これらの仮定の導入は、実際のマルチプロセッサシステムではシステムの構成上の制約などにより、前述した全ての実行効率判定基準1～6を用意するのは難しいと考えることができることから、妥当であると考えられる。

【0088】

[タスクT1の割り付け先の最適化]

<ステップ1-1>タスクT1を読み出す。

<ステップ1-2>タスクT1の直前には仮想的なタスクしかないので、直前は無視できる。

<ステップ1-3>タスクT1の直後にはタスクT2, T3があり、タスクT2, T3はタスクT1とは異なるプロセッサ2, 3にそれぞれ仮割り付けされているので、タスクT1の割り付け先を変更するかどうかの判断に進む。

<ステップ1-4>タスクT1の割り付け先をプロセッサ1からプロセッサ2

， 3に変更することによって、プログラム全体での単位時間当たりのプロセッサ間の通信データ量が小さくなるかどうかを見積もる。

<ステップ1-5>タスクT1をそのままプロセッサ1で実行した場合に必要と予測される実行時間と、割り付け変更候補プロセッサであるプロセッサ2, 3で実行した場合に必要と予測される実行時間を見積もる。

<ステップ1-6>ステップ1-4の結果、割り付け先変更前後でプログラム全体のプロセッサ間の通信データ量は変化することがなく、しかもステップ1-5の結果、タスクT1をプロセッサ1上で実行した方が、必要と予測される実行時間が短いと判明したとする。

<ステップ1-7>ステップ1-6の結果より、タスクT1については割り付け先プロセッサを変更しない旨、決定する。

【0089】

[タスクT2の割り付け先の最適化]

<ステップ2-1>タスクT2を読み出す。

<ステップ2-2>タスクT2の直前には、タスクT1が存在する。

<ステップ2-3>タスクT2の直後にはタスクT3があり、タスクT1, T3はタスクT2とは異なるプロセッサ1, 3にそれぞれ仮割り付けされているので、タスクT2の割り付け先を変更するかどうかの判断に進む。

<ステップ2-4>タスクT2の割り付け先をプロセッサ2からプロセッサ1, 3に変更することによって、プログラム全体での単位時間当たりのプロセッサ間の通信データ量が小さくなるかどうかを見積もる。

<ステップ2-5>タスクT2をそのままプロセッサ2で実行した場合に必要と予測される実行時間と、割り付け変更候補プロセッサであるプロセッサ1, 3で実行した場合に必要と予測される実行時間を見積もる。

<ステップ2-6>ステップ2-4の結果、割り付け変更前後でプログラム全体のプロセッサ間の通信データ量は変化することなく、またステップ2-5の結果、タスクT2をプロセッサ1上で実行した方が、必要と予測される実行時間が最も短いと判明したとする。

<ステップ2-7>ステップ2-6の結果より、タスクT2の割り付け先プロ

セッサをプロセッサ1に変更する旨、決定する。

【0090】

[タスクT3の割り付け先の最適化]

<ステップ3-1>タスクT3を読み出す。

<ステップ3-2>タスクT3の直前には、タスクT1, T2が存在する。

<ステップ3-3>タスクT3の直後にはタスクT7があり、タスクT1, T2はタスクT3とは異なるプロセッサ1に仮割り付けされている。タスクT7はプロセッサ3に仮割り付けされている。タスクT1, T2がプロセッサ1に仮割り付けされているので、タスクT3の割り付け先を変更するかどうかの判断に進む。

<ステップ3-4>タスクT3の割り付け先をプロセッサ1に変更することによって、プログラム全体での単位時間のプロセッサ間の通信データ量が小さくなるかどうかを見積もる。

<ステップ3-5>また、タスクT3をそのままプロセッサ3で実行した場合に必要と予測される実行時間と、割り付け変更先候補プロセッサであるプロセッサ1で実行した場合に必要と予測される実行時間を見積もる。

<ステップ3-6>ステップ3-4の結果、既にタスクT1, T2がプロセッサ1に割り付けられているため、タスクT3の割り付け先をプロセッサ1に変更した方が、プログラム全体のプロセッサ間の通信データ量は減少することが判明したとする。また、ステップ3-5の結果、タスクT3をプロセッサ1上で実行しても必要と予測される実行時間はほとんど変化しないと判明したとする。

<ステップ3-7>ステップ3-6の結果より、タスクT3の割り付け先プロセッサをはプロセッサ1に変更する旨、決定する。

【0091】

[タスクT4の割り付け先の最適化]

<ステップ4-1>タスクT4を読み出す。

<ステップ4-2>タスクT4の直前には仮想的なタスクしかないので、無視できる。

<ステップ4-3>タスクT4の直後にはタスクT6があり、タスクT6はタ

スクT4とは異なるプロセッサ2に仮割り付けされているので、タスクT4の割り付け先を変更するかどうかの判断に進む。

<ステップ4-4>タスクT4の割り付け先をプロセッサ2に変更することによって、プログラム全体での単位時間のプロセッサ間の通信データ量が小さくなるかどうかを見積もる。

<ステップ4-5>タスクT4をそのままプロセッサ3で実行した場合に必要と予測される実行時間と、割り付け変更先候補プロセッサであるプロセッサ2で実行した場合に必要と予測される実行時間を見積もる。

<ステップ4-6>ステップ4-4の結果、タスクT4の割り付け先をプロセッサ2に変更した方が、プログラム全体のプロセッサ間の通信データ量は減少することが判明したとする。また、ステップ4-5の結果、タスクT4をプロセッサ2上で実行しても、必要と予測される実行時間はほとんど変化しないと判明したとする。

<ステップ4-7>ステップ4-6の結果より、タスクT4の割り付け先プロセッサをプロセッサ2に変更する旨、決定する。

【0092】

[タスクT5の割り付け先の最適化]

<ステップ5-1>タスクT5を読み出す。

<ステップ5-2>タスクT5の直前には仮想的なタスクしかないので、無視できる。

<ステップ5-3>タスクT5の直後にはタスクT6があり、タスクT6はタスクT5とは異なるプロセッサ2に仮割り付けされているので、タスクT5の割り付け先を変更するかどうかの判断に進む。

<ステップ5-4>タスクT5の割り付け先をプロセッサ2に変更することによって、プログラム全体での単位時間のプロセッサ間の通信データ量が小さくなるかどうかを見積もる。

<ステップ5-5>また、タスクT5をそのままプロセッサ1で実行した場合に必要と予測される実行時間と、割り付け変更先候補プロセッサであるプロセッサ2で実行した場合に必要と予測される実行時間を見積もる。

<ステップ5-6>ステップ5-4の結果、タスクT5の割り付け先をプロセッサ2に変更した方が、プログラム全体のプロセッサ間の通信データ量は減少することが判明したが、ステップ5-5の結果、タスクT5をプロセッサ2上で実行すると、必要と予測される実行時間が増加すると判明したとする。

<ステップ5-7>ステップ5-6の結果と処理開始前に設定されていた優先度によって、タスクT5の割り付け先プロセッサをプロセッサ2に変更する旨、決定する。

【0093】

[タスクT6の割り付け先の最適化]

<ステップ6-1>タスクT6を読み出す。

<ステップ6-2>タスクT6の直前にはタスクT4, T5があるが、タスクT4, T5は共にタスクT6と同じプロセッサ3に割り付けられているので、無視できる。

<ステップ6-3>タスクT6の直後にはタスクT8があり、タスクT8はタスクT6とは異なるプロセッサ3に仮割り付けされているので、タスクT6の割り付け先を変更するかどうかの判断に進む。

<ステップ6-4>タスクT6をプロセッサ3に割り付け先を変更することによって、プログラム全体での単位時間のプロセッサ間の通信データ量が小さくなるかどうかを見積もる。

<ステップ6-5>タスクT6をそのままプロセッサ2で実行した場合に必要と予測される実行時間と、割り付け変更先候補プロセッサであるプロセッサ3で実行した場合に必要と予測される実行時間を見積もる。

<ステップ6-6>ステップ6-4の結果、タスクT6の割り付け先をプロセッサ3に変更すると、プログラム全体のプロセッサ間の通信データ量は増加することが判明し、ステップ6-5の結果、タスクT6をプロセッサ3上で実行すると、必要と予測される実行時間が増加すると判明したとする。

<ステップ6-7>ステップ6-6の結果から、タスクT6の割り付け先プロセッサを変更しない旨、決定する。

【0094】

[タスクT7の割り付け先の最適化]

<ステップ7-1>タスクT7を読み出す。

<ステップ7-2>タスクT7の直前にはタスクT3があり、タスクT7と異なるプロセッサに割り付けられている。

<ステップ7-3>タスクT7の直後にはタスクT8があり、タスクT8はタスクT7と同じプロセッサ3に割り付けられている。しかし、タスクT7の直前のタスクT3がタスクT7と異なるプロセッサ1に割り付けられているので、タスクT7の割り付け先を変更するかどうかの判断に進む。

<ステップ7-4>タスクT7の割り付け先をプロセッサ1に変更することによって、プログラム全体での単位時間のプロセッサ間の通信データ量が小さくなるかどうかを見積もる。

<ステップ7-5>タスクT7をそのままプロセッサ3で実行した場合に必要と予測される実行時間と、割り付け変更先候補プロセッサであるプロセッサ1で実行した場合に必要と予測される実行時間を見積もる。

<ステップ7-6>ステップ7-4の結果、タスクT7の割り付け先をプロセッサ1に変更すると、プログラム全体のプロセッサ間の通信データ量は増加することが判明し、また、ステップ7-5の結果、タスクT7はプロセッサ1上で実行すると、必要と予測される実行時間が増加すると判明したとする。

<ステップ7-7>ステップ7-6の結果から、タスクT7の割り付け先プロセッサを変更しない旨、決定する。

【0095】**[タスクT8の割り付け先の最適化]**

<ステップ8-1>タスクT8を読み出す。

<ステップ8-2>タスクT8の直前にはタスクT6, T7があり、タスクT6はタスクT8とは異なるプロセッサ3に割り付けられている。

<ステップ8-3>タスクT8の直後にはタスクT9があり、タスクT9はタスクT8とは異なるプロセッサ1に割り付けられているので、タスクT8の割り付け先を変更するかどうかの判断に進む。

<ステップ8-4>タスクT8の割り付け先をプロセッサ1及び2に変更する

ことによって、プログラム全体での単位時間のプロセッサ間の通信データ量が小さくなるかどうかを見積もる。

<ステップ8-5>タスクT8をそのままプロセッサ3で実行した場合に必要と予測される実行時間と、割り付け変更先候補プロセッサであるプロセッサ1及び2で実行した場合に必要と予測される実行時間を見積もる。

<ステップ8-6>ステップ8-4の結果、タスクT8の割り付け先をプロセッサ1または2に割り付けを変更しても、プログラム全体のプロセッサ間の通信データ量は変化しないことが判明し、また、ステップ8-5の結果、タスクT8をそのままプロセッサ3上で実行した方が、必要と予測される実行時間が最も短いと判明したとする。

<ステップ8-7>ステップ8-6の結果から、タスクT8の割り付け先プロセッサを変更しない旨、決定する。

【0096】

[タスクT9の割り付け先の最適化]

<ステップ9-1>タスクT9を読み出す。

<ステップ9-2>タスクT9の直前にはタスクT8があり、タスクT9とは異なるプロセッサ3に割り付けられている。

<ステップ9-3>タスクT9の直後には仮想的なタスクしかないので、無視できる。しかし、タスクT9の直前のタスクT8がタスクT9と異なるプロセッサ3に割り付けられているので、タスクT9の割り付け先を変更するかどうかの判断に進む。

<ステップ9-4>タスクT9をプロセッサ3に割り付け先を変更することによって、プログラム全体での単位時間のプロセッサ間の通信データ量が小さくなるかどうかを見積もる。

<ステップ9-5>タスクT9をそのままプロセッサ1で実行した場合に必要と予測される実行時間と、割り付け変更先候補プロセッサであるプロセッサ3で実行した場合に必要と予測される実行時間を見積もる。

<ステップ9-6>ステップ9-4の結果、タスクT9の割り付け先をプロセッサ1に変更した方が、プログラム全体のプロセッサ間の通信データ量は減少す

ることが判明する。ステップ9-5の結果、タスクT9をプロセッサ3上で実行した方が、必要と予測される実行時間が短いと判明したとする。

<ステップ9-7>ステップ9-6の結果から、タスクT9の割り付け先プロセッサをプロセッサ3へ変更する旨、決定する。

以上のタスク割り付け処理の結果、図10のプログラムについて図11のように仮割り付けされていたタスク群の割り付け先が図12のように最適化される。

【0097】

(割り付け先プロセッサ用のプログラムモジュール取得)

次に、図8中の最適化実行部（割り付け先プロセッサ変更部）26において、変更すべき割り付け先プロセッサ用のプログラムモジュールを取得する処理について説明する。

前述の処理により割り付け先プロセッサが変更されたタスクを実行するためには、割り付け先プロセッサ用のプログラムモジュールを何らかの方法で取得する必要がある。割り付け先が変更されたタスクを実行するためのプログラムを格納したプログラムモジュールは、仮割り付けされたプロセッサが有する命令セットで記述されており、変更された割り付け先プロセッサが有する命令セットとは異なるからである。

【0098】

そこで、本実施形態では例えば図17～図19に示すような3つの手順のいずれかによって、変更された割り付け先プロセッサが有する命令セットで記述された、当該タスクを実行するためのプログラムを格納したプログラムモジュールを取得する。図17～図19は、図13中のステップS13の処理を詳しく示している。

【0099】

図17に示す手順では、対象タスクが元々持っていたプログラムモジュールの記述に用いられている、仮割り付け先プロセッサが有する命令セットに特有の命令を、変更された割り付け先プロセッサの命令セットにおける同じ処理を行う命令に置換することによって、変更された割り付け先プロセッサが有する命令セットで記述されたプログラムモジュールを得る。

【0100】

すなわち、まず割り付け先を変更すべきと判定された対象タスクのプログラムモジュール内の命令が割り付け先プロセッサに存在しない命令か否かを判定する（ステップS301）。ステップS301の判定結果がYESの場合には、その命令を割り付け先プロセッサ用の同一処理を行う命令に置換することによって、割り付け先プロセッサ用のプログラムモジュールを生成する（ステップS302）。ステップS301の判定結果がNOの場合には、新たなプログラムモジュールの取得は必要がないため、処理を終了する。ステップS301～S302の処理をステップS303で全ての命令について処理が終了したと判断されるまで行う。

【0101】

図18は、図17のステップS302に代わる処理を示している。この手順では、対象タスクが元々持っていたプログラムモジュールのソースコードから、変更された割り付け先プロセッサが有する命令セットで記述されたプログラムモジュールを生成することのできるコンパイラを用いて、変更された割り付け先プロセッサが有する命令セットで記述されたプログラムモジュールを取得する。

【0102】

図19は、同様に図17のステップS302に代わる処理を示している。この手順では、変更された割り付け先プロセッサが有する命令セットで記述された、当該タスクのプログラムモジュールをファイルシステム中から、もしくはネットワークから検索して取得する。

【0103】

（タスク割り付け処理手順2）

次に、本実施形態に基づくタスク割り付けの処理手順の他の例について説明する。図20は、タスク割り付け処理手順2の流れを示している。

図13に示したタスク割り付け処理手順1では、プログラムを構成する全タスクを各プロセッサに対して、仮割り付け（ステップS11）、割り付け先プロセッサの変更によりプログラム実行効率が向上するか否かの判定（ステップS12）及び割り付け先プロセッサの変更によりプログラム実行効率が向上すると判定

された対象タスクに対する割り付け先プロセッサの変更（ステップS13）を順次行っている。

【0104】

これに対し、図20に示すタスク割り付け処理手順2では、まずプログラムを構成する全タスクから一つのタスクを選択し（ステップS21）、選択したタスクについて図13中のステップS11～S13に相当する処理を行う（ステップS22～S24）。そして、プログラムを構成する全タスクについてプロセッサに対する割り付け処理が終了したと判断されるまで、ステップS21～S24の処理を繰り返す。

【0105】

このようにして図20の処理が終了すると、プログラムを構成する全タスクがそれぞれ適切なプロセッサに割り付けられるので、マルチプロセッサシステムは当該プログラムを効率よく実行することができる。

【0106】

（タスク割り付け処理手順3）

図21は、本実施形態に基づくもう一つのタスク割り付け処理手順の流れを示している。このタスク割り付け処理手順3では、図13のステップS11に相当するプログラムを構成する全タスクの仮割り付けを行った後、プログラムの実行を開始する（ステップS31～S32）。この後、プログラムの実行途中でステップS33において所定の条件が満たされた場合にのみ、図13中のステップS12～S13に相当する処理を行う（ステップS34～S35）。そして、ステップS36でプログラムの実行が終了したと判断されるまで、ステップS32～S35の処理を繰り返す。

【0107】

ここで、ステップS33における「所定の条件」としては、例えば以下の条件が挙げられる。

[条件1] 一定時間間隔で訪れるシステムタイマによる割り込みがあった。

[条件2] あるプロセッサから、過負荷になりそうだという通知があった。

[条件3] アイドル状態にあるプロセッサからの割り込みがあった。

[条件4] あるプロセッサが入出力命令を発行したことにより、入出力命令の実行完了待ち状態に入ったという通知があった。

[条件5] あるプロセッサから、一つのタスクの実行を終了したという通知があった。

ただし、これらの条件1～5はあくまで例であり、この限りではない。

【0108】

(プログラムモジュール複合体)

次に、本発明の他の実施形態を説明する。

これまでの説明では、本実施形態のヘテロマルチプロセッサシステムが実行すべきプログラムとして、タスクとタスク間の依存関係で記述されたプログラムであって、しかも例えば図10に示したように各タスクが特定のプロセッサ用の命令セットで記述されたプログラムモジュールのみで構成されている例について説明した。

【0109】

ヘテロマルチプロセッサシステムが実行対象とするプログラムは、それを構成する全タスクが一つのプログラムモジュールとして与えられている必要は必ずしもない。プログラムを構成する全タスクのうち、少なくとも一つのタスクは、二つ以上の異種プロセッサがそれぞれ有する命令セットによって記述された複数のプログラムモジュールを含む複合体（これをプログラムモジュール複合体という）であってもよい。

【0110】

例えば、図22(a)に示すプログラムモジュール複合体40Aは、命令セットA, B, Cでそれぞれ記述されているプログラムモジュール41, 42, 43を含んでいる。図22(b)に示すプログラムモジュール複合体40Bは、命令セットA, Bでそれぞれ記述されているプログラムモジュール41, 42を含んでいる。

【0111】

プログラムを構成する各タスクは、例えばタスクの内容やタスクの作成者の意図に応じて図22(a) (b)に示されるようにいずれかのプログラムモジュー

ル複合体として与えられるか、あるいは図22(c)に示されるように一つのプログラムモジュール41のみとして与えられる。

【0112】

プログラムを構成する全タスクが、共通の複数の命令セットでそれぞれ記述されている複数のプログラムモジュールを含むプログラムモジュール複合体として与えられてもよい。すなわち、プログラムを構成する全タスクが、いずれも例えば図22(a)のようなプログラムモジュール複合体であってもよい。

【0113】

上述のようなプログラムモジュール複合体という構造をタスクに適用した場合には、図15の処理により割り付け先変更の対象となるタスクを選定して割り付け先プロセッサを変更するかどうかの判定を行う際に、前述した実行効率判定基準の他に「割り付け先プロセッサの命令セットで記述されたプログラムモジュールが、当該タスクのプログラムモジュール複合体の中に存在する」という判定基準を設けて、これを必ず満たさなければならない基準とすることが望ましい。これは、割り付け変更先候補プロセッサの命令セットで記述されたプログラムモジュールがプログラムモジュール複合体の中に存在しない限り、割り付け先を変更しても変更された割り付け先プロセッサ上で当該タスクを実行することはできないからである。

【0114】

次に、プログラムを構成する少なくとも一部のタスクが上述したプログラムモジュール複合体である場合の図13のステップS11及びS31の処理について説明する。

図23は、図13中のステップS11の本実施形態に対応する処理の詳細を示す。割り付けるべき対象タスクのプログラムモジュール複合体中のプログラムモジュールを記述している命令セットが何であるかを判断し(ステップS111)、その命令セットを有するプロセッサに対して対象タスクを割り付ける(ステップS112)。

【0115】

次に、図24～図26を用いて図13中のステップS13の本実施形態に対応

する処理手順の種々の例について述べる。

図24の処理手順では、まず図13中のステップS12で決定された割り付け先プロセッサは、対象タスクのプログラムモジュール複合体に含まれるプログラムモジュールの命令セットのいずれかを用いるプロセッサであるかどうかを判定する（ステップS311）。ステップS311の判定の結果がYESであれば、そのプログラムモジュール複合体から当該命令セットで記述されたプログラムモジュールを取得する（ステップS312）。

【0116】

一方、ステップS311の判定の結果がNOであれば、対象タスクのプログラムモジュール複合体に含まれる任意の一つのプログラムモジュールを選択する（ステップS313）。次いで、図17中のステップS302と同様に、ステップS313で選択された命令セットで記述されたタスクのプログラムモジュール中の命令を、割り付け先プロセッサ用の当該命令と同一処理を行う命令に置換することによって、割り付け先プロセッサ用のプログラムモジュールを生成する（ステップS314）。

【0117】

図25の処理手順では、ステップS321～S323の処理については図24中のステップS311～S312と全く同様であり、ステップS324の処理だけが異なっている。ステップS321の判定の結果がNOの場合には、対象タスクのプログラムモジュール複合体に含まれる任意の一つのプログラムモジュールをステップS323で選択する。

【0118】

次に、図18に示した処理と同様に、ステップS323で選択されたプログラムモジュールのソースコードから、変更された割り付け先プロセッサが有する命令セットで記述されたプログラムモジュールを生成することのできるコンパイラを用いて、変更された割り付け先プロセッサが有する命令セットで記述されたプログラムモジュールを生成する。

【0119】

図26の処理手順では、ステップS331, S333の処理については図24

中のステップS311, S312と全く同様であり、ステップS334の処理だけが異なっている。すなわち、ステップS331の判定結果がNOの場合にはステップS334に移り、割り付け先プロセッサが有する命令セットで記述された、当該タスクのプログラムモジュールをファイルシステム中から、もしくはネットワークから検索して取得する。

このように、プログラムを構成するタスクがプログラムモジュール複合体で構成される場合にも、本発明によるタスク割り付けは有効である。

【0120】

【発明の効果】

以上説明したように、本発明によれば命令セットが異なる複数のプロセッサから構成されるヘテロなマルチプロセッサシステムにおいて、記述に用いられている命令セットが異なる複数のタスク群を実行する際に、よりプログラムの実行効率が向上するような、命令セットの異なるプロセッサへの割り付け先の変更を実現することが可能となり、それによってシステム全体の実行効率を大きく改善することができる。

【図面の簡単な説明】

【図1】本発明の一実施形態に係るマルチプロセッサシステムの構成を示すブロック図

【図2】同実施形態におけるタスク割り付けプログラムの第1の実装例を示す図

【図3】同実施形態におけるタスク割り付けプログラムの第2の実装例を示す図

【図4】同実施形態におけるタスク割り付けプログラムの第3の実装例を示す図

【図5】同実施形態におけるタスク割り付けプログラムの第4の実装例を示す図

【図6】マルチプロセッサシステムで実行されるタスクとタスク間の依存関係で記述されたプログラムの例を示す図

【図7】タスクの実行の様子の種々の例を示す図

【図8】同実施形態におけるタスク割り付けシステムの機能的構成を示すブロック図

【図9】図9中の最適化実行判定部25の詳細な構成を示すブロック図

【図10】異なる複数の命令セットで記述されたプログラムモジュールで構成されるタスクとタスク間の依存関係で記述されたプログラムの例を示す図

【図11】

図10のプログラムをプログラムモジュールの記述に用いられている命令セットを基準として各プロセッサに割り付けた例を示す図

【図12】図11の割り付け例を仮割り付けとして本実施形態に従って割り付け先を変更した後の割り付け例を示す図

【図13】同実施形態におけるタスク割り付け処理の一例を示すフローチャート

【図14】図13中の仮割り付け処理の一例を示すフローチャート

【図15】図13中の判定処理の一例を示すフローチャート

【図16】図15の判定処理の前処理の一例を示す図

【図17】図13中の割り付け先プロセッサ変更処理の一例を示すフローチャート

【図18】図13中の割り付け先プロセッサ変更処理の他の例を示すフローチャート

【図19】図13中の割り付け先プロセッサ変更処理の別の例を示すフローチャート

【図20】同実施形態におけるタスク割り付け処理の他の例を示すフローチャート

【図21】同実施形態におけるタスク割り付け処理の別の例を示すフローチャート

【図22】本発明の他の実施形態におけるタスク割り付け処理に関わるモジュール複合体についての説明図

【図23】同実施形態における仮割り付け処理の一例を示すフローチャート

【図24】同実施形態における割り付け先プロセッサ変更処理の一例を示す

フローチャート

【図25】同実施形態における割り付け先プロセッサ変更処理の他の例を示すフローチャート

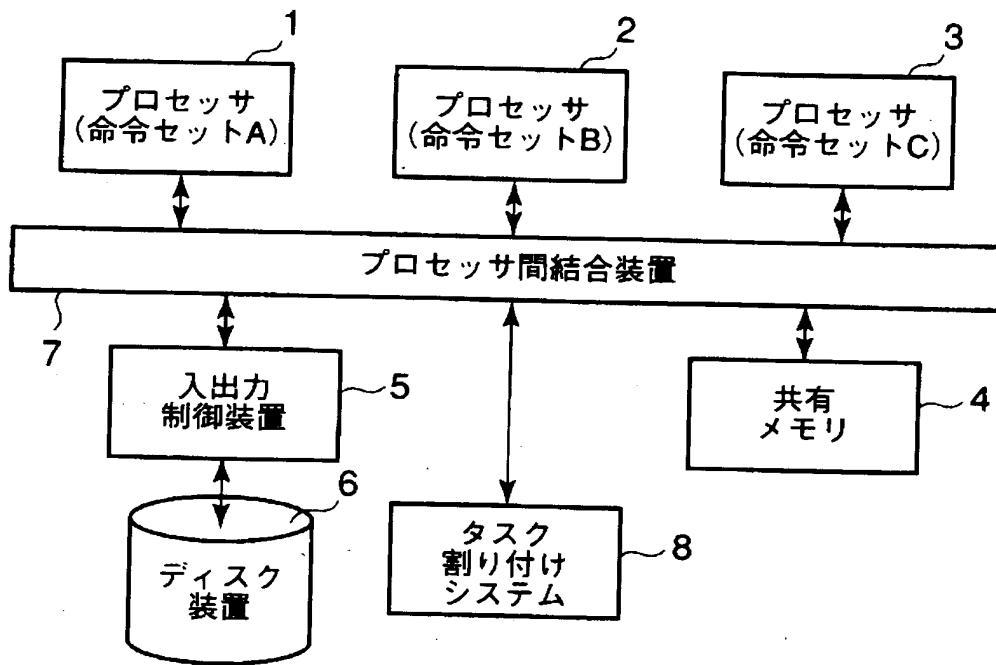
【図26】同実施形態における割り付け先プロセッサ変更処理の別の例を示すフローチャート

【符号の説明】

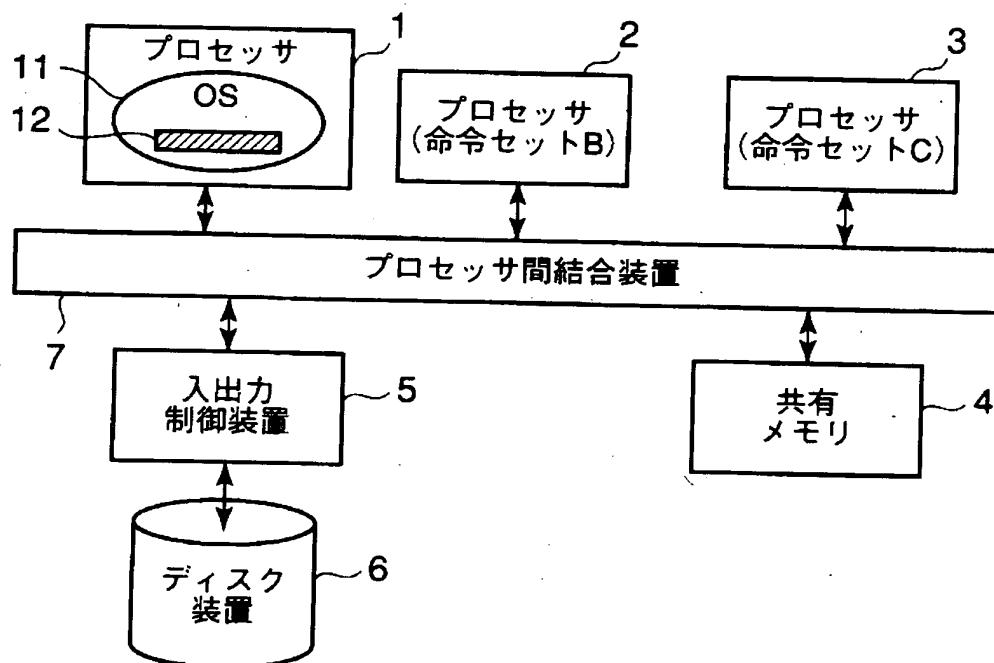
- 1～3 … プロセッサ
- 4 … 共有メモリ
- 5 … 入出力制御装置
- 6 … ディスク装置
- 7 … プロセッサ間結合装置
- 8 … タスク割り付けシステム
- 9 … 管理用プロセッサ
- 11, 13 … オペレーティングシステム
- 12 … タスク割り付けプログラム

【書類名】 図面

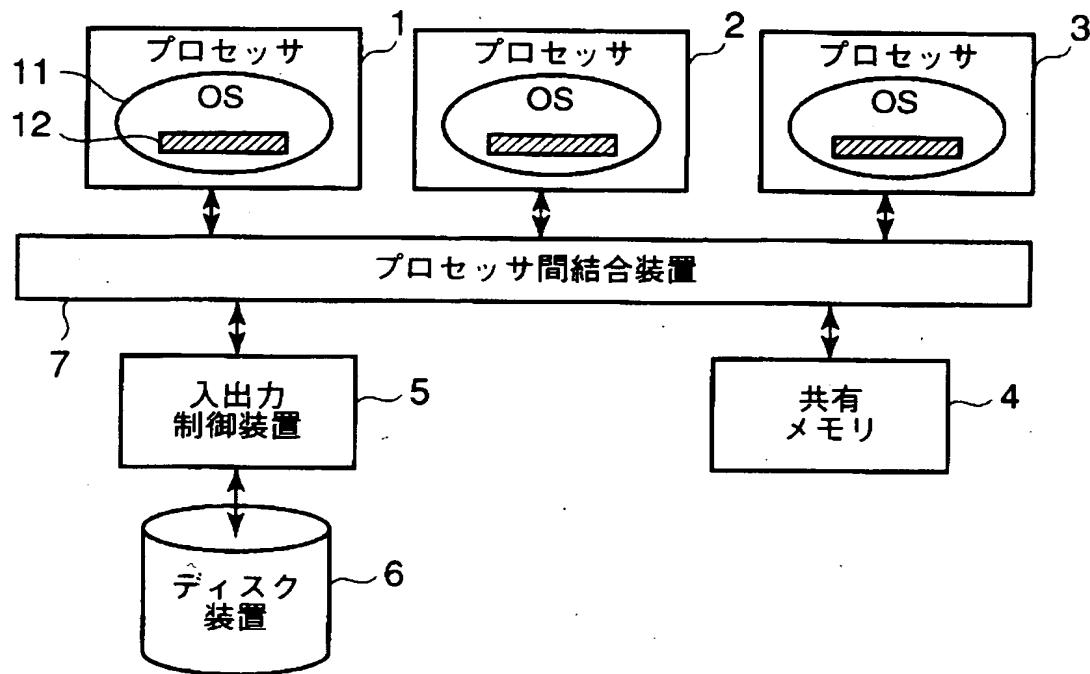
【図 1】



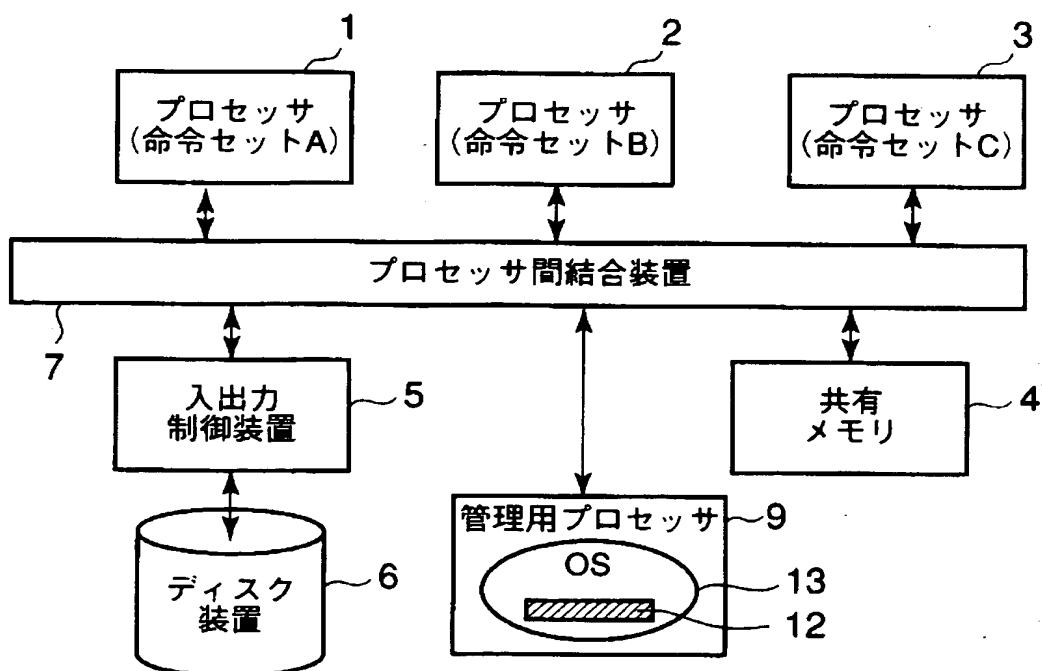
【図 2】



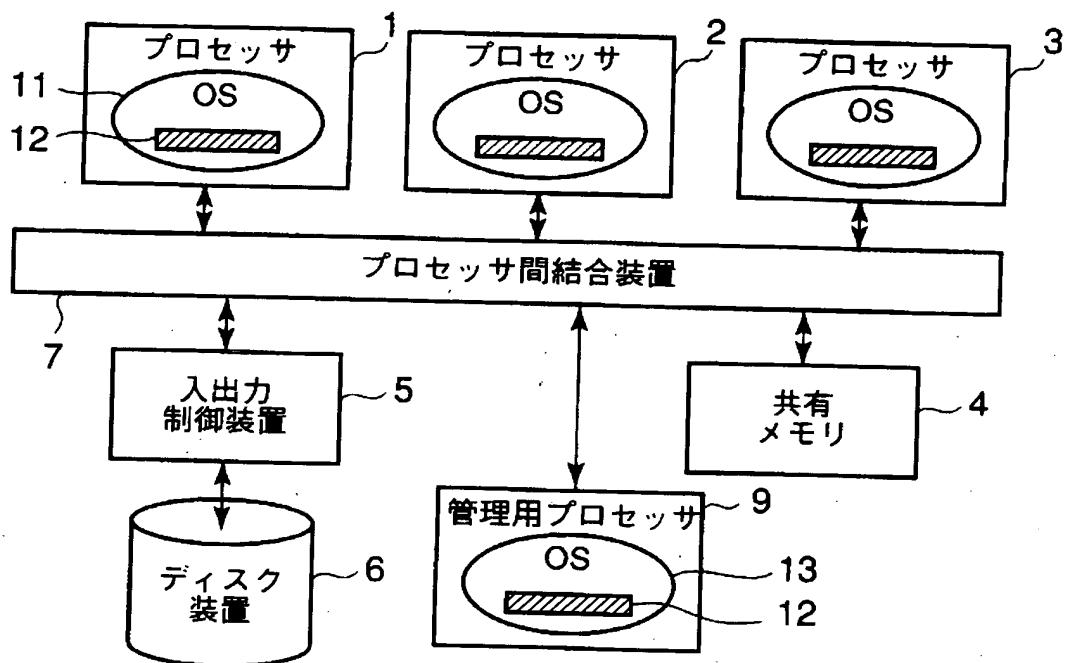
【図 3】



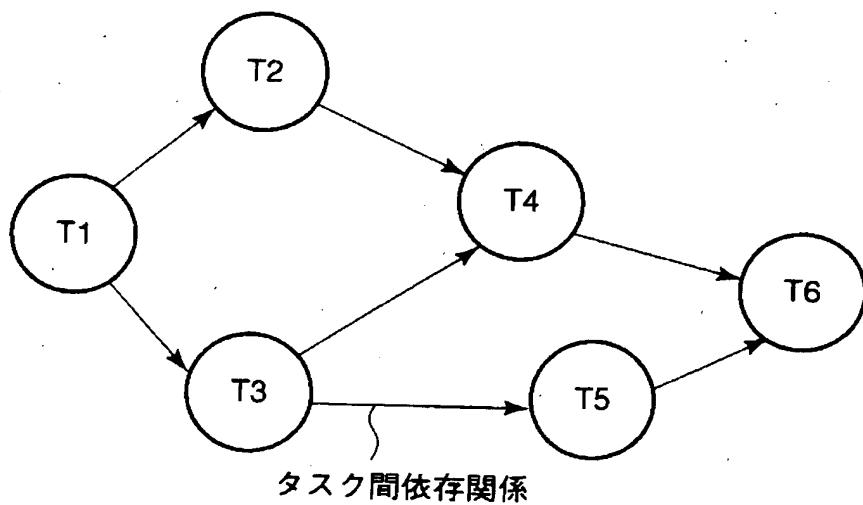
【図 4】



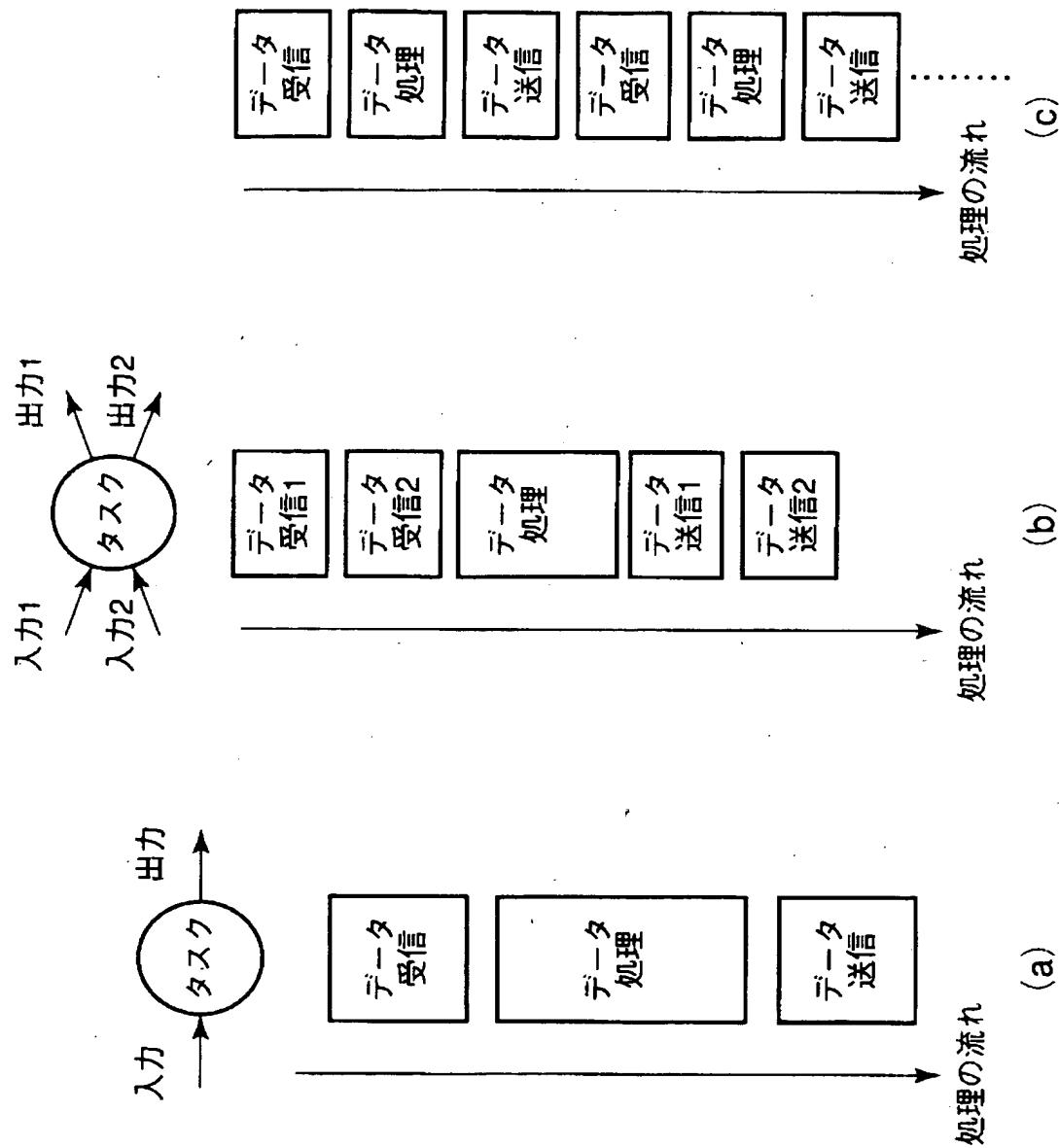
【図 5】



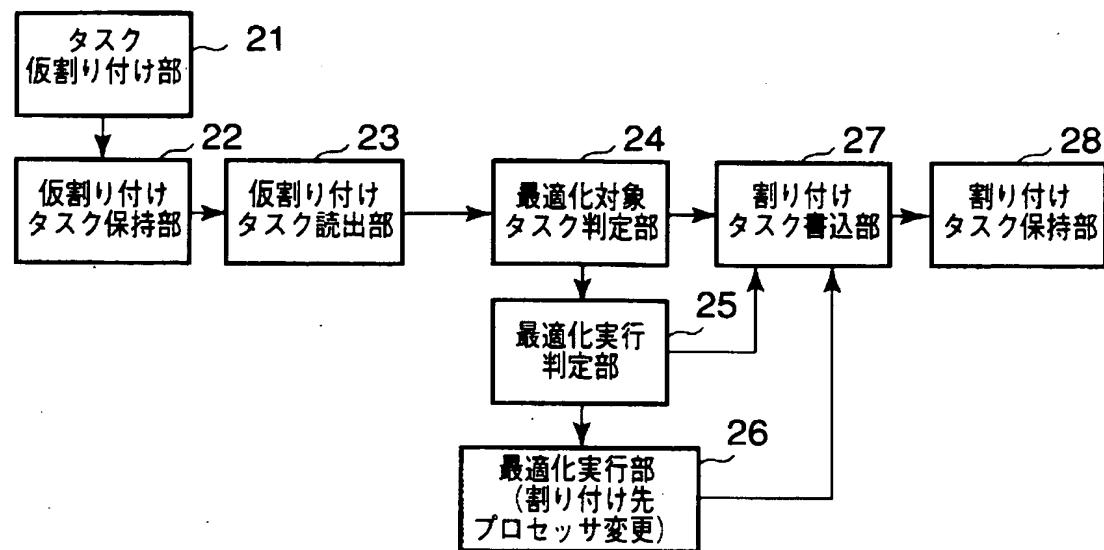
【図 6】



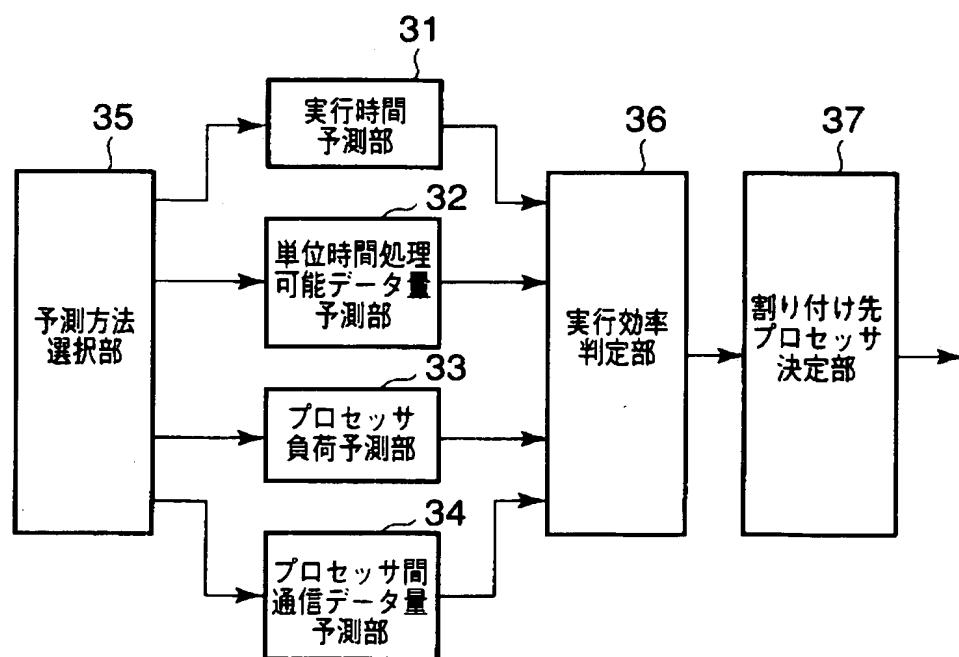
【図7】



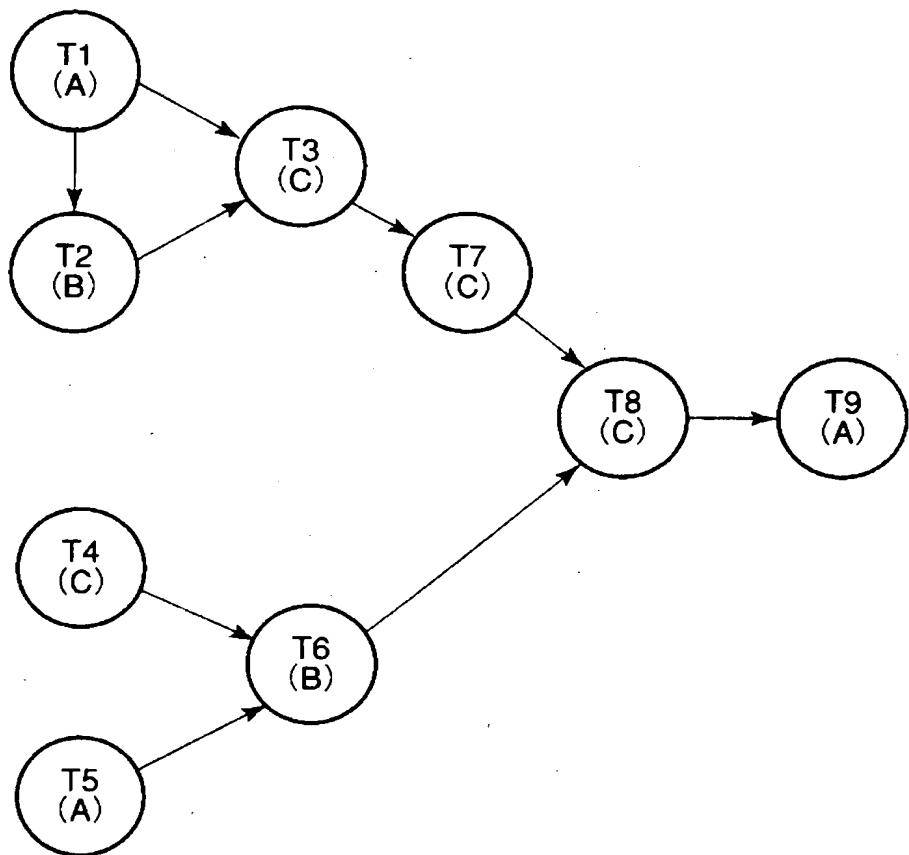
【図8】



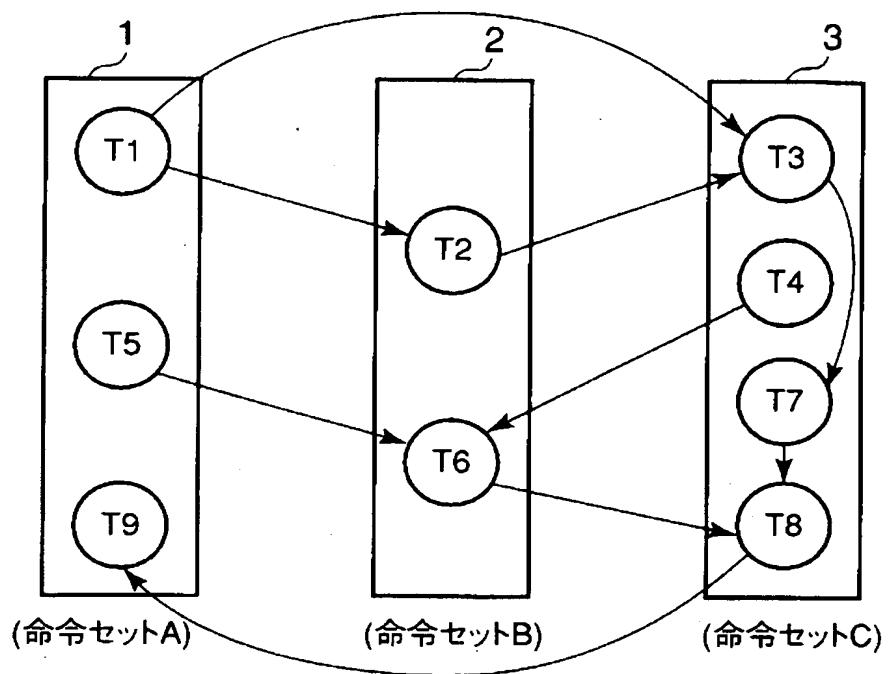
【図9】



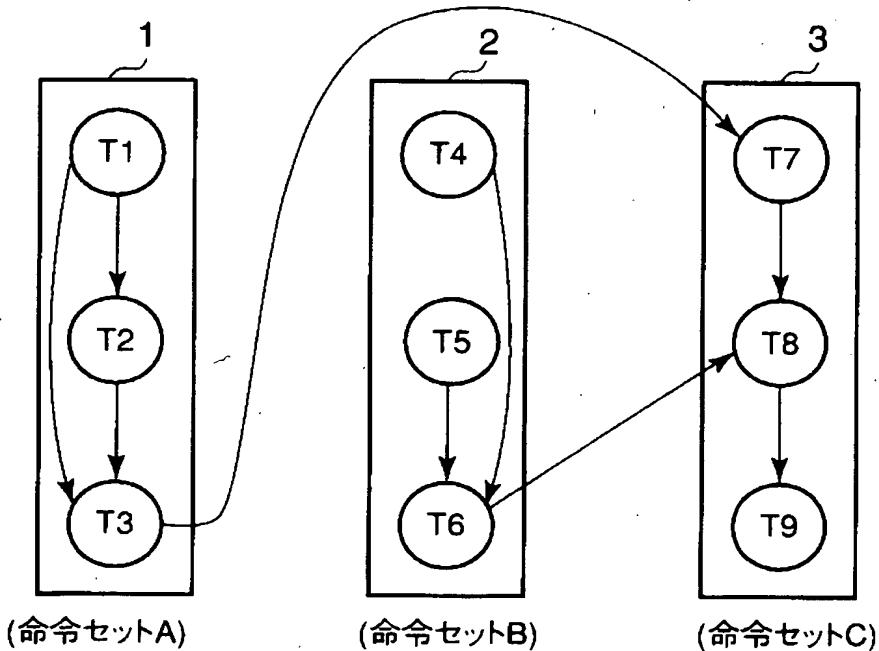
【図10】



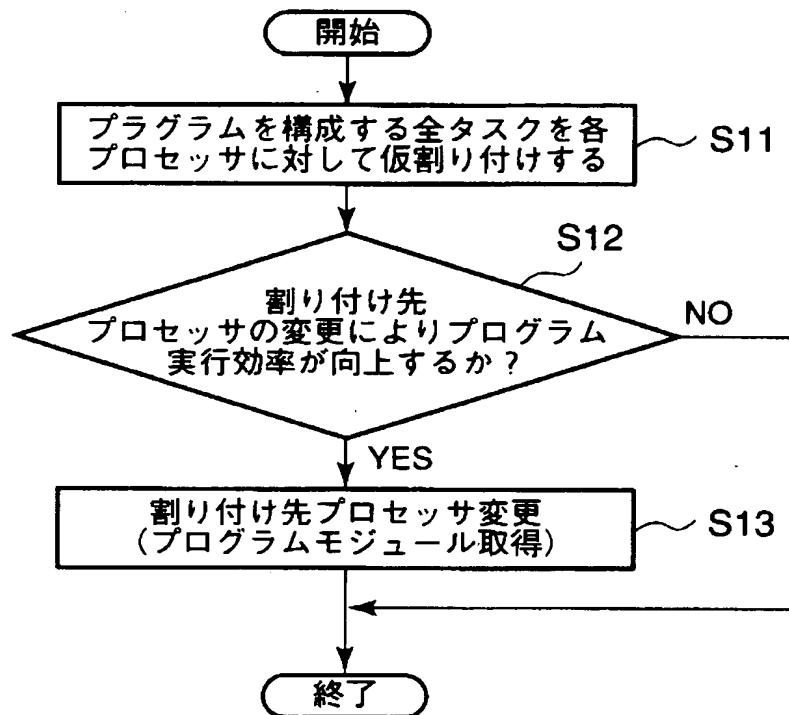
【図 1 1】



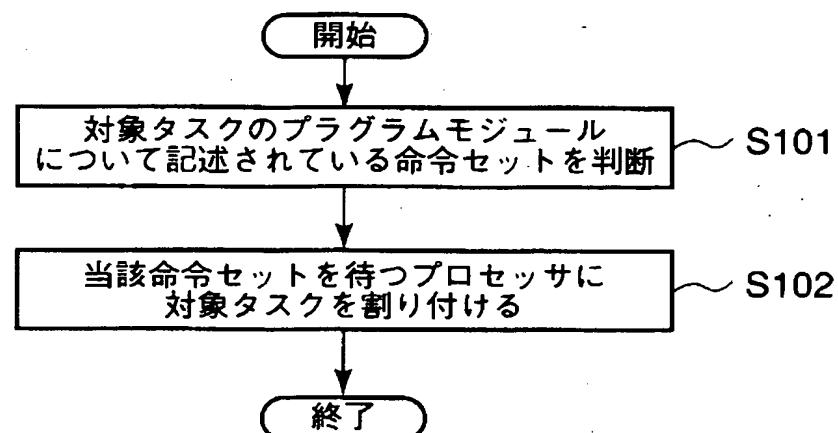
【図 1 2】



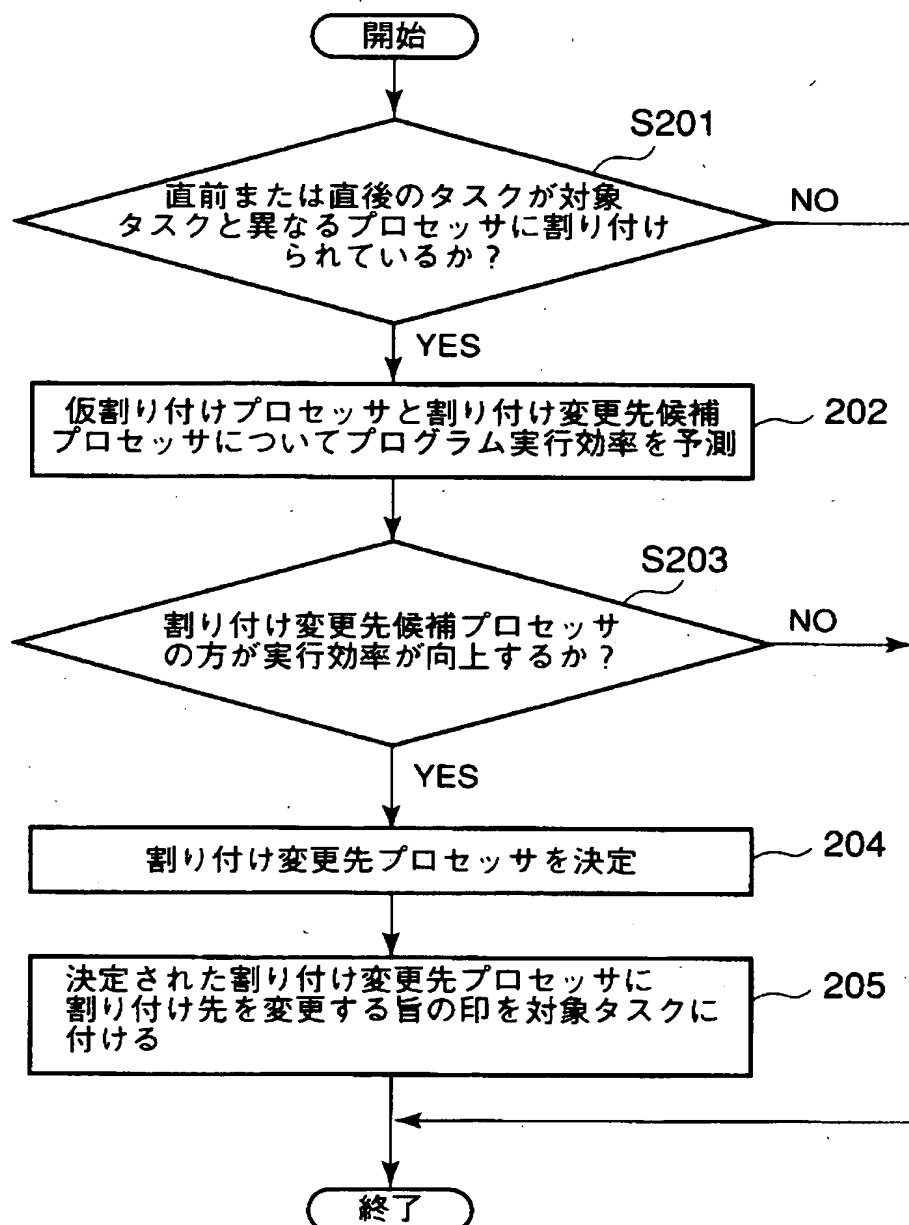
【図13】



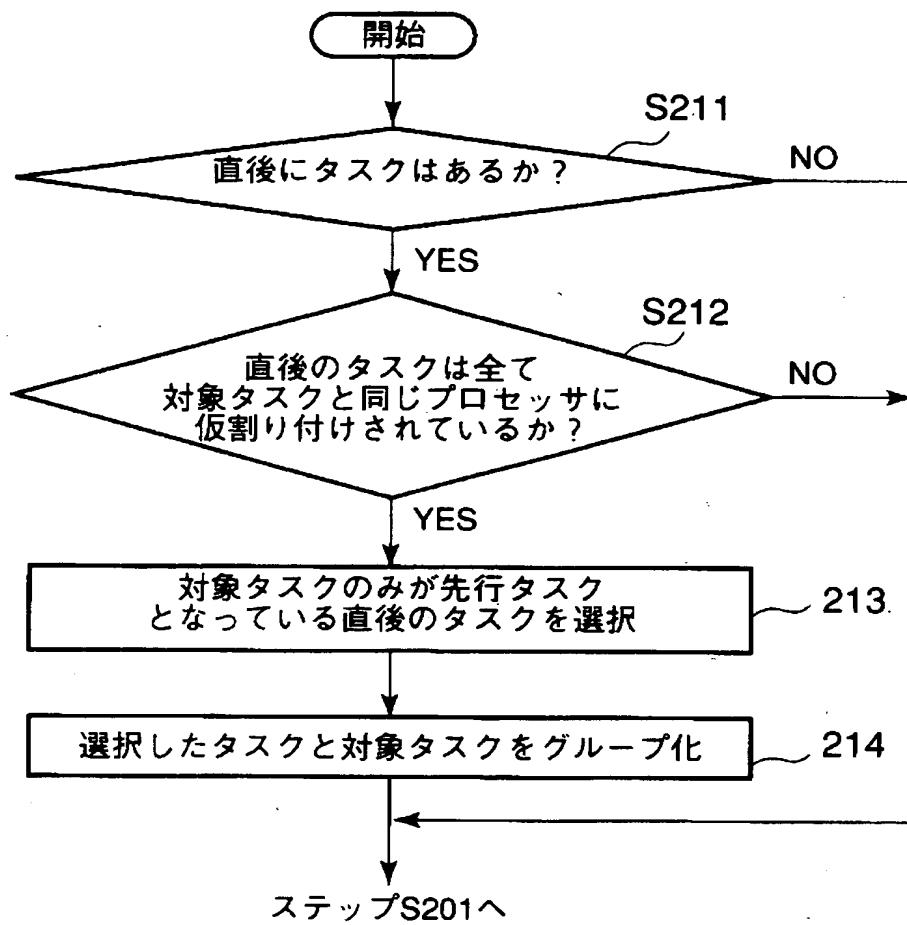
【図14】



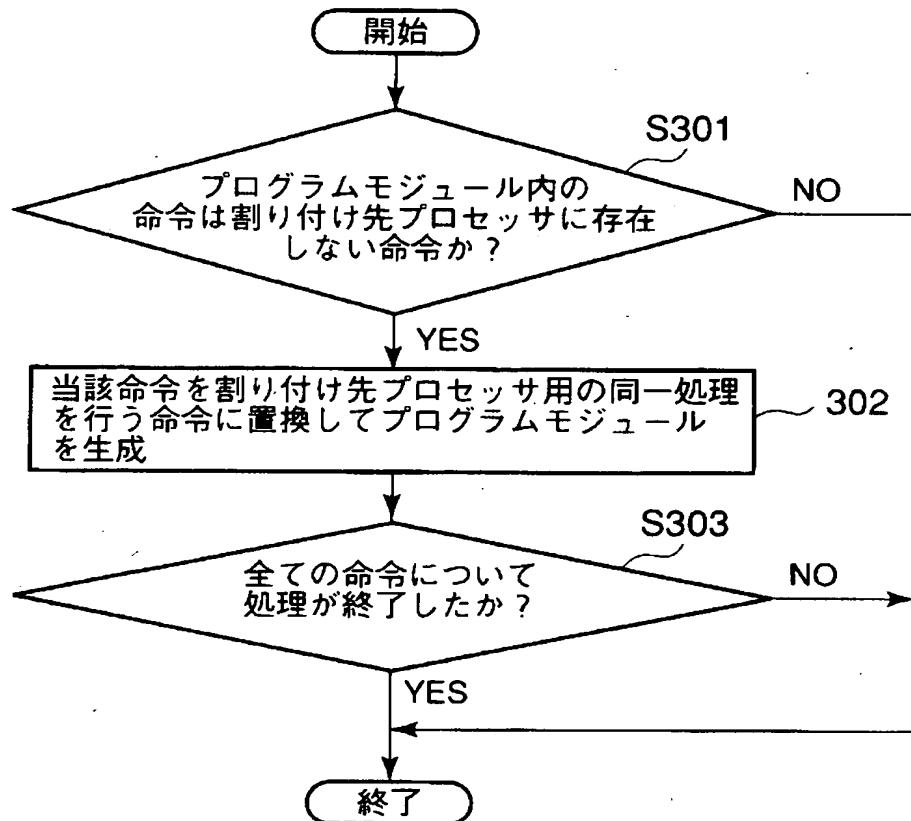
【図15】



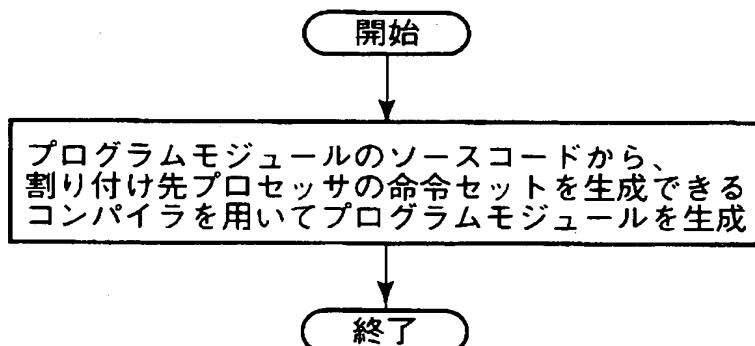
【図16】



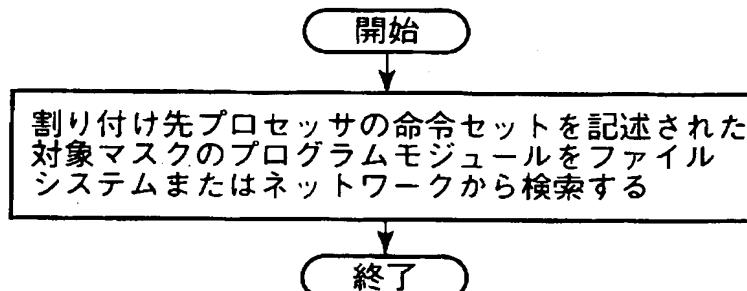
【図17】



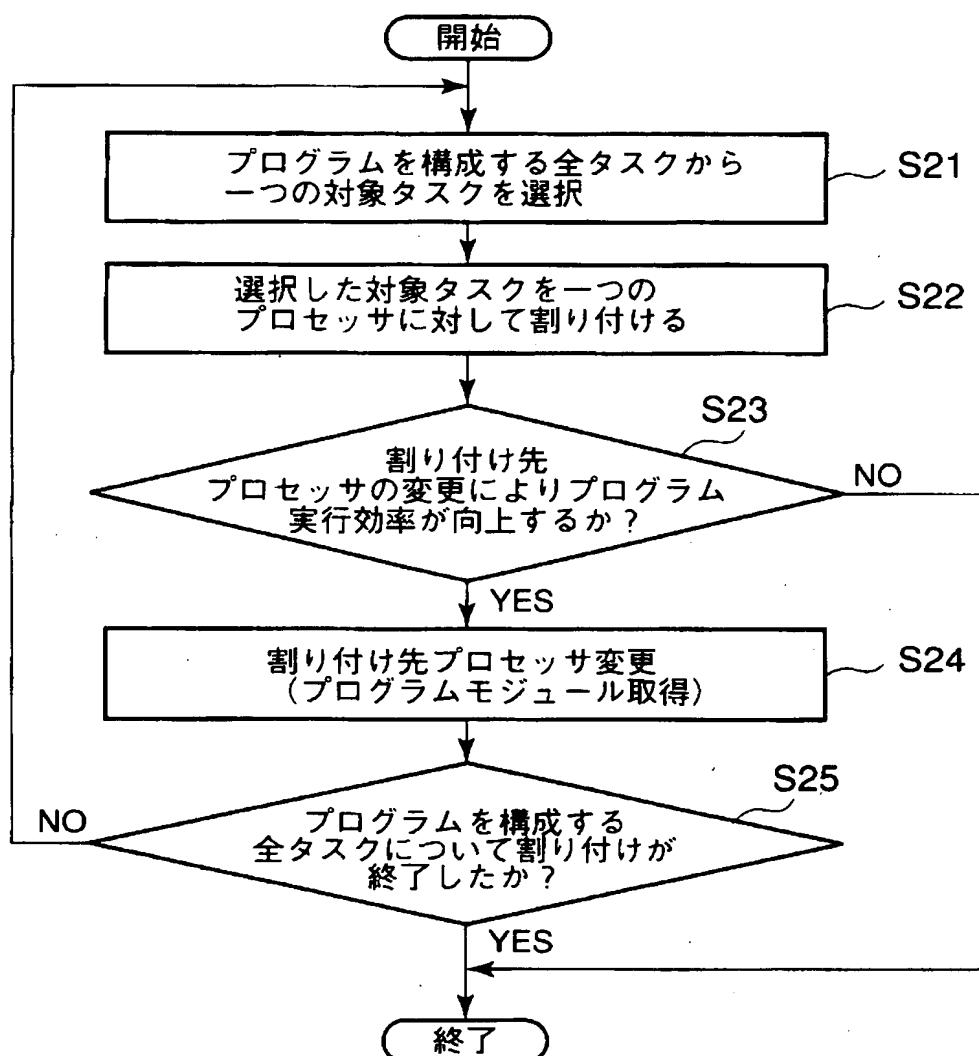
【図18】



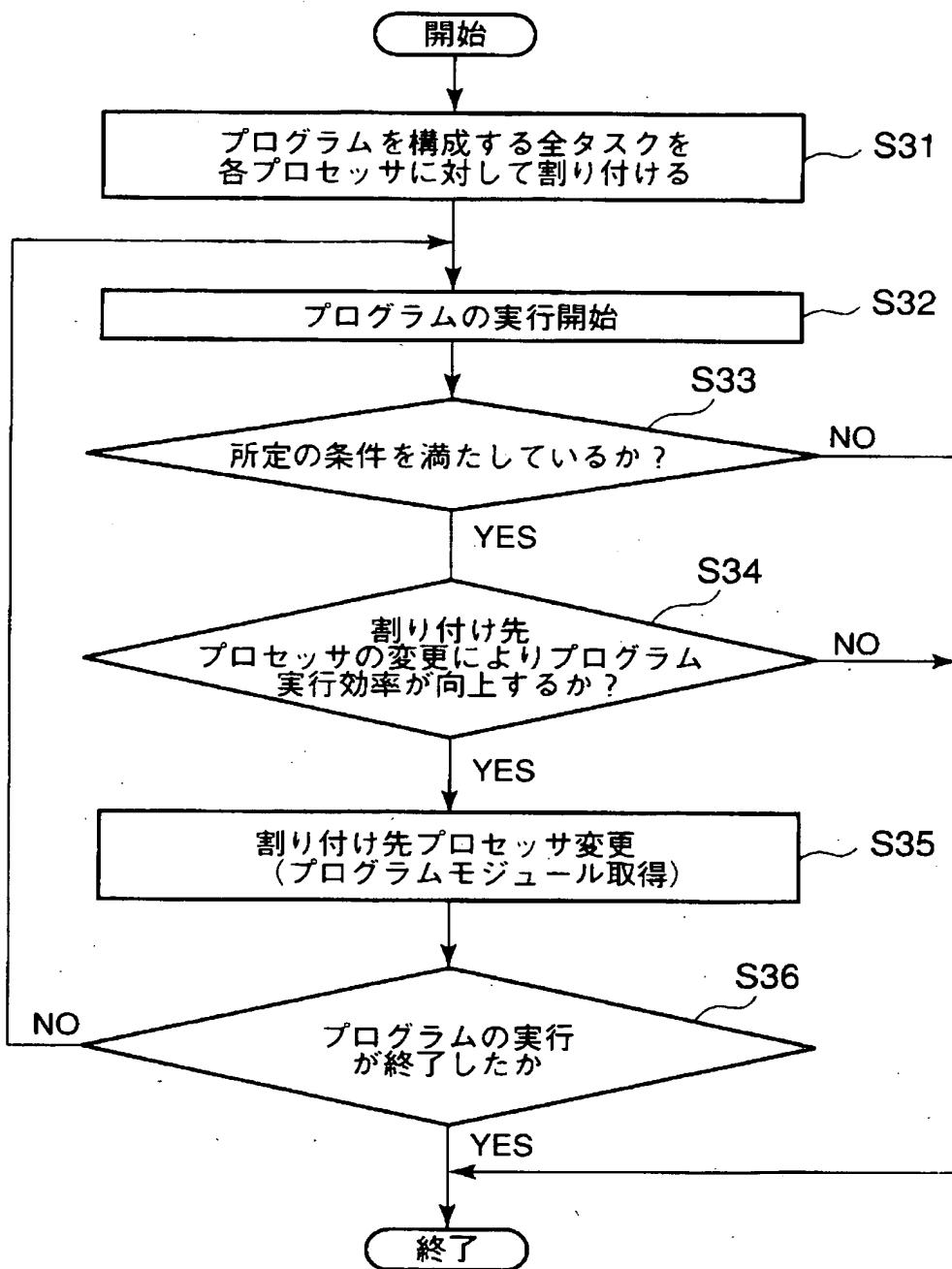
【図 19】



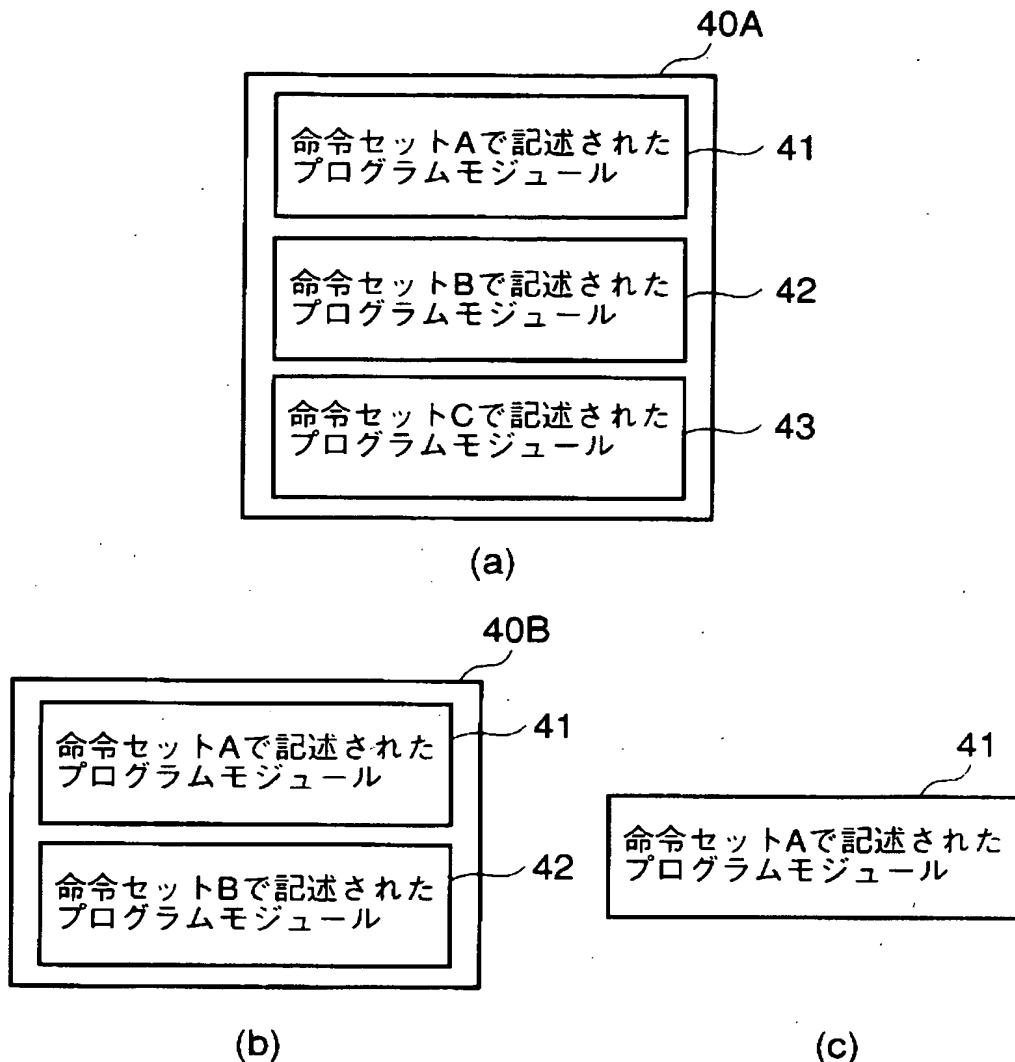
【図 20】



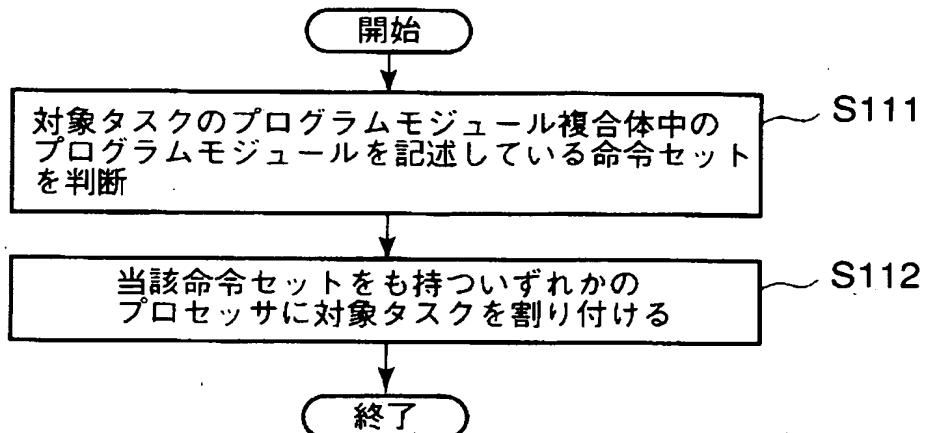
【図21】



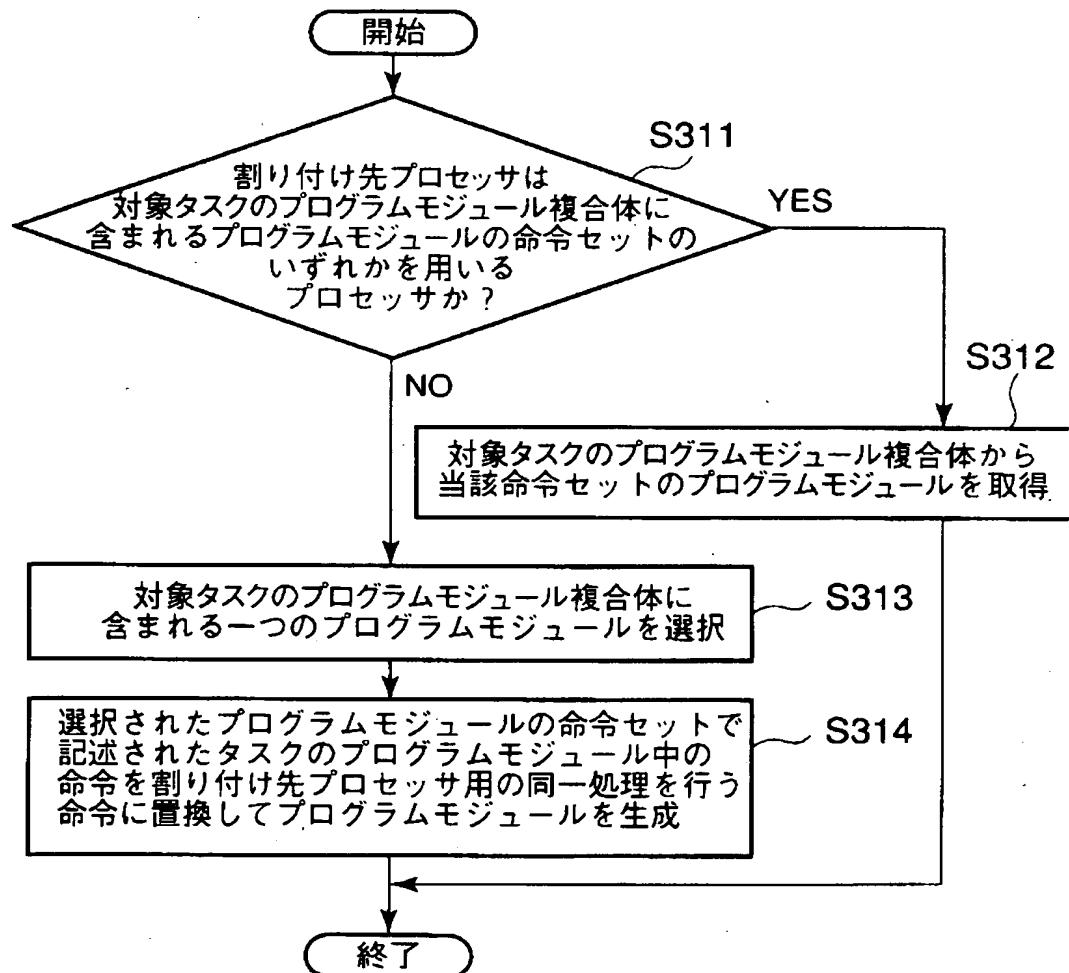
【図22】



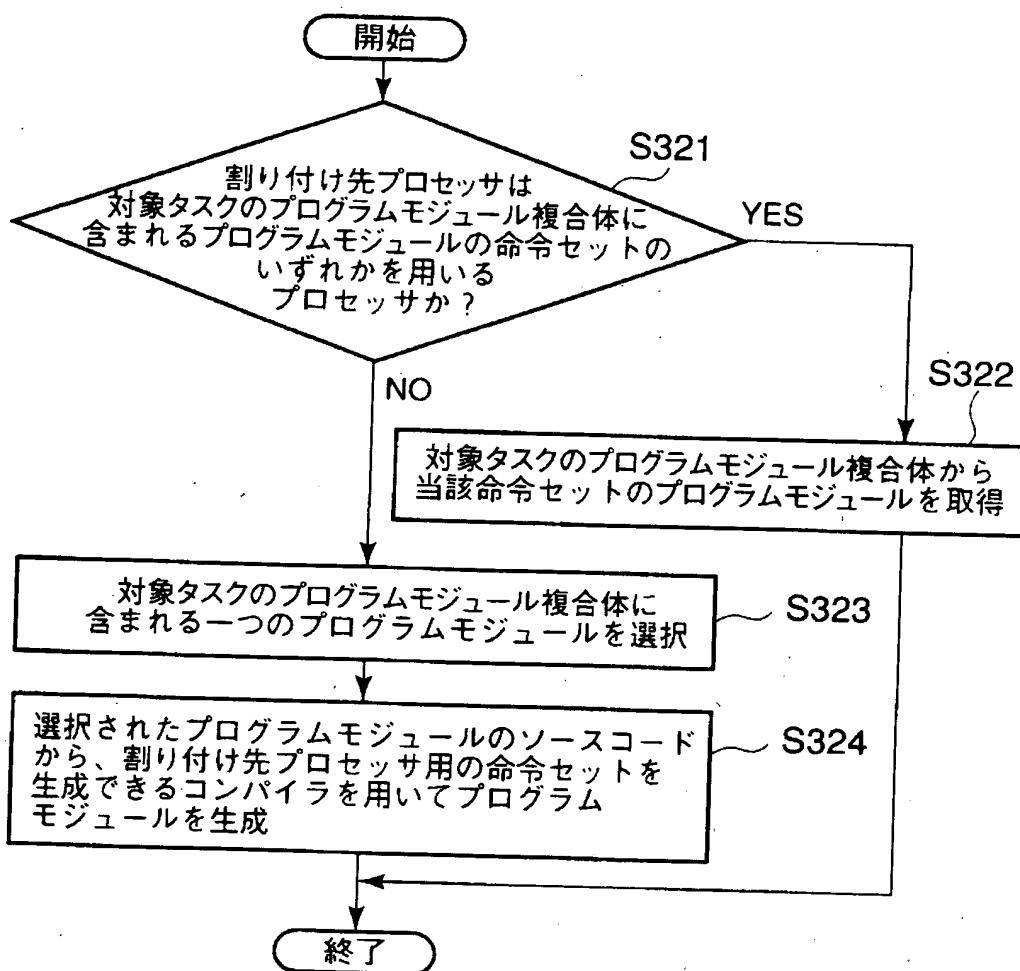
【図23】



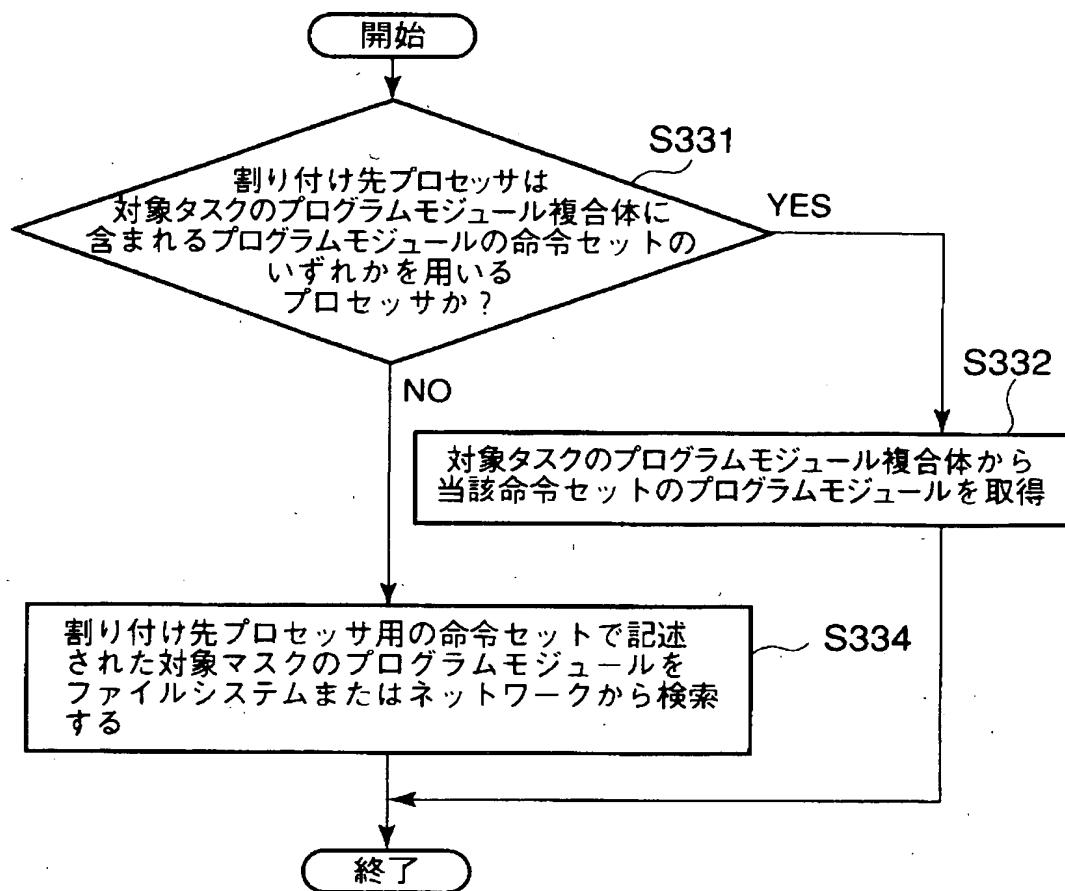
【図24】



【図25】



【図26】



【書類名】 要約書

【要約】

【課題】 命令セットの異なる異種プロセッサを有するマルチプロセッサシステムにおけるプログラムの実行効率を向上させるタスク割り付け方法を提供する。

【解決手段】 命令セットの異なる複数の異種プロセッサを含むマルチプロセッサシステムによって実行されるプログラムを構成する、該命令セットのいずれかでそれぞれ記述された複数のタスクをプロセッサに割り付ける際に、各タスクのうちの少なくとも一つの対象タスクを記述している命令セットを有する第1プロセッサに対して対象タスクを割り付け（S11）、対象タスクの割り付け先を第1プロセッサと異なる第2プロセッサに変更することによって、プログラムの実行効率が向上するか否かを判定し（S12）、実行効率が向上する対象タスクについては、第2プロセッサが有する命令セットによって記述されたプログラムモジュールを取得して割り付け先を第2プロセッサに変更する（S13）。

【選択図】 図13

特願2002-335632

出願人履歴情報

識別番号 [000003078]

1. 変更年月日 2001年 7月 2日

[変更理由] 住所変更

住 所 東京都港区芝浦一丁目1番1号
氏 名 株式会社東芝

2. 変更年月日 2003年 5月 9日

[変更理由] 名称変更

住 所 東京都港区芝浦一丁目1番1号
氏 名 株式会社東芝